

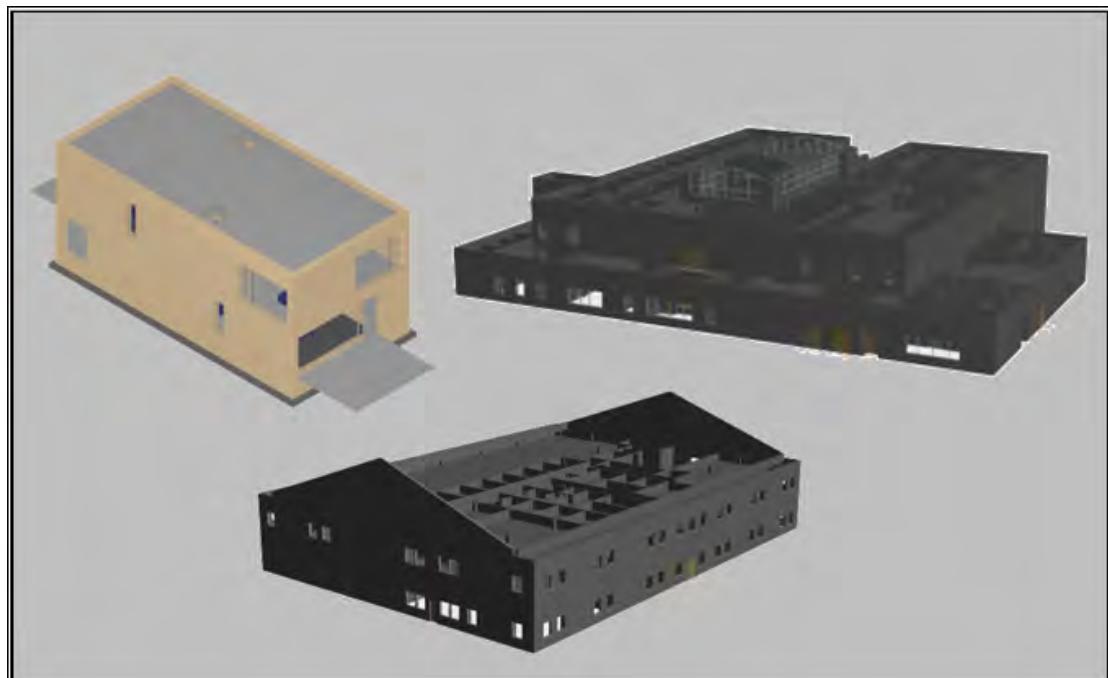


**US Army Corps
of Engineers®**
Engineer Research and
Development Center

Experimental Building Information Models

Mark Johnson and Kristine K. Fallon

September 2011



Experimental Building Information Models

Mark Johnson and Kristine K. Fallon

Kristine Fallon Associates, Inc.
11 E. Adams Street, Suite 1100
Chicago, IL 601603

Final report

Approved for public release; distribution is unlimited.

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under Contract W912HZ-D-0003, "Experimental Building Information Models"

Monitored by Construction Engineering Research Laboratory
U.S. Army Engineer Research and Development Center
2902 Newmark Drive, Champaign, IL 61822

Abstract: In a previous Army research project, a core life-cycle building information model (BIM) was developed based on example models for three specific buildings: a Duplex Apartment, a Clinic, and an Office. These models were developed inconsistently over time by different modelers and contain a various levels of detail and quality of content across disciplines. This report documents the development and creation of three building information models that include information for architectural, structural, plumbing, electrical, heating, and ventilating. The three buildings are a family housing unit, a small office building, and a medical clinic. The authors developed a set of electronic building components, or common object library, and necessary setup protocols, and created a model-development plan to provide guidance to modelers. These guidelines may be helpful to designers and BIM managers to successfully create models that comply with Industry Foundation Class (IFC) based standards such as the Facility Management Handover Model View Definition or Construction Operations Building information exchange (COBie) format.

Table of Contents

List of Tables.....	vi
Preface	vii
1 Introduction.....	1
1.1 Background.....	1
1.2 Objectives	1
1.3 Approach	1
1.4 Development of Common Object Library	2
1.5 Identification and Resolution of Limitations in COTS Functionality.....	4
1.6 Model Development	5
1.7 Quality Control	7
1.8 BIM Execution Planning	7
1.9 Scope	10
2 Common Object Library Description	11
2.1 Objectives	11
2.2 Approach	11
2.3 Scope	11
2.4 Modeling Standards	11
2.5 COBie Properties	13
2.6 Family Parameters and Project Parameters	14
2.7 Shared Parameters	15
2.8 OmniClass Parameters.....	15
2.9 Instance Parameters and Type Parameters.....	16
2.10 IFC Export Override Parameters	16
2.11 Mapping of Revit Parameters to COBie Spreadsheet.....	17
2.11.1 Facility Tab.....	18
2.11.2 Floor Tab.....	20
2.11.3 Space Tab.....	21
2.11.4 Type Tab	22
2.11.5 Component Tab	24
2.11.6 Attribute Tab.....	25
2.11.7 Zone Tab.....	26
2.11.8 System Tab.....	27
2.12 Common Object Library Files	27
2.13 Revit Component Families	28
2.14 Revit System Families	32
2.15 Revit Template Files.....	35
2.16 IFC File Export.....	35
2.17 IFC Export Settings	36
2.18 IFC Export Override.....	36

2.19	COBie File Conversion	38
2.20	COBie File Issues Review	42
2.21	Common Object Library IFC Files.....	43
3	Duplex Apartment Model Description	45
3.1	Objectives	45
3.2	Approach.....	45
3.3	Scope	45
3.4	Modeling Standards.....	45
3.5	File Origin	46
3.6	File Setup	46
3.6.1	<i>Reference Model</i>	46
3.6.2	<i>Architectural Model</i>	46
3.6.3	<i>MEP Model</i>	46
3.7	IFC File Export.....	47
3.8	COBie File Conversion	47
3.9	COBie File Issues Review	50
3.10	Manual Review of COBie Files	50
3.11	Architectural Model Review.....	51
3.12	MEP Model Review	51
4	Clinic Model Description	53
4.1	Objectives	53
4.2	Approach.....	53
4.3	Scope	53
4.4	Modeling Standards	53
4.5	File Origin	55
4.6	File Setup	55
4.6.1	<i>Reference Model</i>	55
4.6.2	<i>Structural Model</i>	55
4.6.3	<i>Architectural Model</i>	56
4.6.4	<i>MEP Model</i>	57
4.7	MEP Spaces and Rooms.....	57
4.8	IFC File Export.....	58
4.9	IFC Model Partitions.....	58
4.10	Limitations of Model Partitions.....	59
4.11	Export IFC Using “Current View Only” Option	60
4.12	COBie File Conversion	60
4.13	COBie File Issues Review	64
4.14	Manual Review of COBie Files	64
4.15	Structural Model Review	65
4.16	Architectural Model Review.....	65
4.17	MEP Model Review	66
5	Office Model Description.....	68
5.1	Objectives	68
5.2	Approach	68

5.3	Scope	68
5.4	Modeling Standards	68
5.5	File Origin	69
5.6	File Setup	70
5.6.1	<i>Reference Model</i>	70
5.6.2	<i>Structural Model</i>	70
5.6.3	<i>Architectural Model</i>	71
5.6.4	<i>MEP Model</i>	71
5.7	IFC File Export.....	72
5.8	IFC Model Partitions.....	72
5.9	Limitations of Model Partitions	73
5.10	Export IFC Using “Current View Only” Option	73
5.11	COBie File Conversion	74
5.12	COBie File Issues Review	77
5.13	Manual Review of COBie Files	78
5.14	Structural Model Review	78
5.15	Architectural Model Review.....	79
5.16	MEP Model Review	79

Report Documentation Page

List of Tables

Table 2-1: Project File Name.....	12
Table 2-2: Project Name.....	12
Table 2-3: Discipline Abbreviations.....	12
Table 2-4: Family File Names.....	13
Table 2-5: Facility Tab Parameters.....	18
Table 2-6: Floor Tab Parameters.....	20
Table 2-7: Space Tab Parameters.....	21
Table 2-8: Type Tab Parameters.....	22
Table 2-9: Component Tab Parameters.....	24
Table 2-10: Attribute Tab Parameters.....	25
Table 2-11: Zone Tab Parameters.....	26
Table 2-12: Component Family File Names.....	28
Table 2-13: System Family Names.....	32
Table 2-14: Revit Template File Names.....	35
Table 2-15: IFC Export Override Parameter Settings.....	37
Table 2-16: IFC Export Categories.....	39
Table 2-17: Component Families with no COBie Type.....	39
Table 2-18: Component Families with Modified Category Settings.....	40
Table 2-19: Curtain Wall IFC Categories.....	41
Table 2-20: BimServices Duplicate Data Fields.....	42
Table 2-21: Files not Processed by BimServices Issues Check.....	43
Table 3-1: Duplex Apartment BIM Files	45
Table 3-2: Revit Families with no IFC Type.....	49
Table 3-3: MEP System Families with Multiple COBie Types.....	52
Table 4-1: Clinic BIM Files	54
Table 4-2: Clinic MEP Model Partitions	54
Table 4-3: Revit Families with no IFC Type.....	61
Table 4-4: MEP System Families with Multiple COBie Types.....	67
Table 5-1: Office BIM Files	69
Table 5-2: Office MEP Model Partitions	69
Table 5-3: Revit Families with no IFC Type.....	75
Table 5-4: MEP System Families with Multiple COBie Types.....	80

Preface

This study was performed by Eyak Technology, LLC, for the US Army Engineer Research and Development Center – Construction Engineering Research Laboratory (ERDC-CERL) under Contract W912HZ-D-0003, “Experimental Building Information Models.” The technical monitor was Dr. E. William East.

The work was supervised by the Engineering Processes Branch (CF-N) of the Facilities Division (CF), U.S. Army Engineer Research and Development Center – Construction Engineering Research Laboratory (ERDC-CERL). At the time of publication, Donald K. Hicks was Chief, CEERD-CF-N; L. Michael Golish was Chief, CEERD-CX; and Martin J. Savoie, CEERD-CV, was the Technical Director for Installations. The Deputy Director of ERDC-CERL was Dr. Kirankumar Topudurti and the Director was Dr. Ilker Adiguzel.

COL Kevin J. Wilson was the Commander and Executive Director of ERDC, and Dr. Jeffery P. Holland was the Director.

1 Introduction

1.1 Background

To enforce contractual requirements for the delivery of information exchanges based on Industry Foundation Class (IFC) based standards, such as the Facility Management Handover Model View Definition or the Construction Operations Building information exchange (COBie) format, it is necessary that application software be capable of meeting such specifications and that use of that software comply with those requirements. To date there have been no controlled experiments to create building information models (BIMs) that fully document all steps needed to achieve a specified outcome. Research that requires BIM data also needs controlled experimental models in order that the results of the work may be repeated and extended. Such models do not currently exist.

1.2 Objectives

The primary objective of this project is to develop, document, and create three (3) building information models that include the specified information for architectural, structural, plumbing, electrical, heating, and ventilating as described in Contract W912HZ-D-0003, "Experimental Building Information Models." The secondary objective of this project is to document the method used to create these models to assist later modelers to create similar models.

1.3 Approach

To meet the first objective, we are developing a set of electronic building components (common object library) that conform to relevant standards. These components will then be assembled to create the required models. The steps, standards and tools used in this process will be described in a separate chapter of this report.

To meet the second objective, we have created a model development plan, documented in this chapter to provide guidance to future modelers. In addition to general guidance that will be relevant to any BIM authoring software we are also providing the common object library in Revit 2011 format that includes both Revit families and templates.

The software tools used on this project are:

- Revit Architecture 2011: BIM authoring application for Architectural design, will be used to develop the architectural and structural models.
- Revit MEP 2011: BIM authoring application for MEP system design, will be used to develop the mechanical, electrical, plumbing, and fire protection disciplines.
- Navisworks Manage 2011: BIM collaboration tool used for model review and clash detection.
- BimServices v2010-12-28: BIM utility for compliance checking and conversion between IFC 2x3, IFCXML 2x3, and COBie v2.40 formats.
- ProjNet COBie: Compliance tool used to verify the COBie v2.40 deliverable.

1.4 Development of Common Object Library

The first step in developing the common object library is to review the existing models of each building to identify all object types that will be required to recreate each model under a common standard. Each government-supplied model will be reviewed using schedules to list all of the model objects by category. This list must be reviewed and condensed to eliminate duplicates. Additionally, since the government-supplied models have been created to different levels of detail and do not include all of the requested systems, objects and systems not included in the government-supplied models will be added to the list.

Once the list of required objects has been completed, the actual model objects must be gathered and organized to form the common object library. For the purpose of this project, model objects will be collected or created for use in Revit 2011, from the following sources in order of preference:

- **Revit 2011 Content Library:** Most objects will be taken directly from the default content library that comes with Revit 2011.
- **Free online libraries:** Objects not available in the default library can often be found online at websites dedicated to the model authoring tools. In this case, the primary online source will be www.seek.autodesk.com, a content search engine that allows users to search for content from multiple online sources for use in AutoDesk software such as Revit. It is important to verify that none of the content incorporated in the government models is use-restricted.

- **Custom creation:** If the required objects are not available in a free public library, they will be created in Revit Architecture or Revit MEP 2011 for use on this project.

In addition to the individual model objects, template files must also be developed in the common object library. In a Revit project, the templates contain project standards such as line weights, but also contain a number of unique, hard-coded model elements such as walls, floors, and roofs. These objects, known as “system families” in Revit, will be added directly to the templates for use on this project. A separate template will be developed for each discipline being modeled: architectural, structural, mechanical, electrical, plumbing and fire protection.

When selecting model objects for the common object library, they must be reviewed to verify that their geometric content (3D model geometry) and non-geometric content (data properties) are a sufficient representation of the real world object. Since many of the available objects are developed by different sources and for different purposes, they can vary in quality and content. For the purpose of this project, the selected objects will be generic representations of common building elements with properties and functionality consistent with the basic use of the Revit software. This will ensure that the end result can be reproduced by other project teams without the need for specialized model content.

After the model objects have been selected for the common object library, it is necessary to extend their property sets to include data fields for the required COBie properties. There are two basic methods of adding these parameters within Revit:

- “Family parameters” can be added to the individual model files. This is the most common method of adding properties to objects in Revit, but it must be done to each individual file.
- “Project parameters” can be added to object categories within the template files. The properties are not carried by the individual model element files, but are instead added automatically when they are inserted into the project template.

For the purpose of this project, the “project parameter” approach will be used to add COBie data parameters to the templates. This method will

allow users to develop object libraries without the need to map special properties to each individual model object.

Once the common object library has been established, one copy of each object will be exported to IFC and to COBie formats, and to verify compliance with the project requirements each object will be verified using the BimServices tool.

1.5 Identification and Resolution of Limitations in COTS Functionality

Since Autodesk does not provide a COBie template for Revit, , it is necessary to create one. We have identified a number of gaps in content supplied with Revit that will need to be addressed to meet the COBie requirements. These gaps, and the proposed solutions, are:

COBie data: Native Revit elements do not include all of the parameters needed to carry much of the required COBie data. The appropriate parameters will be added to elements used to create the Revit model. Project parameters will be added to the object categories in the template files to store the required COBie data.

IFC export: Revit's IFC export routine includes user settings to map object categories to IFC format, but does not include any user settings to map individual parameters to IFC properties. Revit will export the parameters as IFCPROPERTYSINGLEVALUE attached to each object, using the same parameter name as defined within Revit. In order for the BimServices tool to transform the IFC properties to the COBie spreadsheet, they must be added to the Revit files using the correct naming standards.

IFC Import: When importing an IFC file, Revit will load the object geometry and data properties contained in the IFC file, but there is still some loss of functionality due to the file conversion. The resulting objects are not always fully functional Revit objects. For example, Revit's native ability to parametrically control the width of a door opening is not available if the door was imported from an IFC file. In order for an IFC common object library to be used in Revit, individual objects would be required for each size: a 36" door, a 30" door, and so forth. For the purpose of this project, only native Revit elements will be used to develop the models.

BimServices Transform1 utility: The BimServices tool is still under development, with the current release being v2010-12-28. This version includes properties required for COBie v2.40, however some of the fields are not correctly populated from the IFC files. Details on the limitations encountered will be detailed in subsequent chapters of this report.

1.6 Model Development

Building models for each of the 3 buildings will be developed to a level consistent with the “Coordinated Design” or “60% Design” phase, and will include objects that would typically be shown in drawings at a scale of $\frac{1}{4}$ ” = 1'-0”. While the model content will be developed to the coordinated design phase, the associated documentation and full engineering design of the building systems is not required in the scope of this project.

Using the common object library with custom COBie parameters developed above, and using the government supplied models for reference, building information models will be developed for each of the required buildings. Each of the BIMs will include separate models for architectural, structural, plumbing, electrical, heating and ventilating systems. The individual models for each building will be linked together to create the composite BIM.

When creating a composite BIM from multiple models, it is necessary to define a common origin point to ensure that the models register with each other within the 3D space. Revit uses 2 separate coordinate systems to allow the user to locate buildings within the 3D space:

- **Project Internal:** This coordinate system is the global coordinate system within each individual file. This system defines the 0,0,0 origin point in a conventional X,Y,Z coordinate system.
- **Shared Coordinates:** This coordinate system creates an alternate origin point that can be shared among multiple files, regardless of the origin of each individual file. This coordinate system can be used to register multiple buildings together on the same site, regardless of each building’s project internal origin.

The project internal coordinate system will be used to establish a consistent origin point for each individual file in the composite BIM. The

“0,0,0” point of the project internal coordinate system will define the intersection of column grid A/1 at the first floor in each building project.

It should be noted that on larger projects, the composite BIM may be comprised of multiple models within each discipline. For example, a high rise project could be divided into groups of floors rather than modeled as an entire discipline in one file. This allows the project team to keep the file sizes smaller, making it easier to work on large projects. Consistent use of the file origin is critical to keeping all of the different files consistently referenced into the composite BIM.

In order to use the existing IFC models as reference, the government-supplied files will be imported into Revit. During this file conversion process, there may be some loss of data or functionality, so the imported files will be checked against the original IFC file using other software such as Navisworks Manage 2011, as noted under the Quality Control section below.

The 3D model imported from the IFC file will be used as a background to aid in the development of a new, native Revit model. The imported model will be saved as a separate RVT project file, and linked into the main project file using the Revit file linking tools. This link can later be removed.

In some cases, it may also help to have a 2D drawing extracted from the 3D model, such as when laying out a large floor plan. In such cases, a Revit plan view can be exported as a 2D Autocad DWG file using Revit’s export function. This 2D DWG file can then be referenced back into Revit using the file linking tools to be used as a 2D background, and unloaded when the model is complete.

At a minimum, the model content for each discipline will include the following:

- **Architectural:** Walls, doors, windows, roof, floors, ceilings, rooms/spaces.
- **Structural:** Foundations, floor slabs, framing, stairs and elevators.
- **Mechanical:** Heating, ventilating, and air conditioning equipment, thermostats, ducts, and piping.
- **Plumbing:** Equipment, fixtures, drains, and piping.

- **Fire Protection:** Equipment, piping, and devices.
- **Electrical:** Lighting, power and data outlets, disconnects, fixtures, fire alarm and notification devices, emergency lighting and fixtures, and communications equipment.

Objects that will not be modeled at this level of development include:

- Detailed architectural elements such as trim, baseboard, crown molding.
- Structural connections, bridging, rebar, framed openings, miscellaneous metals.
- Small mechanical components such as sensors, hangers, flanges.
- Small plumbing and fire protection components such as clean outs, hangers.
- Small electrical components such as hangers, wiring.

The government-provided models were developed to varying levels of completeness and quality. Where design information is incomplete or missing, assumptions consistent with common design practices will be made to fill in the design gaps and develop the model and to complete the systems as outlined above.

1.7 Quality Control

The native model files will be reviewed using room schedules and equipment schedules to verify the required model content. Schedules will also be exported and used as reference to verify the completeness of the IFC and COBie export.

IFC files will be reviewed in Navisworks to verify the geometric model export.

COBie files will be converted to IFCXML using the BimServices Transform1 tool and then checked for compliance using the Transform1 Issues tool. The results of the BimServices compliance check will be compared to the results of the ProjNet-COBie tool.

1.8 BIM Execution Planning

A major promise of Building Information Modeling is to create a computable building description. The first evidence of this computability

is the ability to automatically extract plans, sections and elevations from the model within the same program. The next step is to submit the model to some kind of analysis – structural, energy, or other. In other words, a BIM created by one software program should be directly readable by another software program. This interoperability is frequently achieved within a group of products from a single vendor or via customized API-based interfaces. However, there are limitations to any vendor's product and partner offerings. Invariably, there arises a need to use software from an unrelated vendor for some special purpose. A common reason is that different companies working on the same project have different software preferences.

A major purpose of a BIM Execution Plan is to identify and make reliable the information exchanges required either during project execution (design and/or construction) or for the entire facility life cycle. Although the facility life cycle is often referenced, current practice tends to focus primarily on assigning responsibility for creating the models of specific building systems and elements and secondarily on how dimensionally precise those models are. This information, combined with data format preference, is typically used to define “exchange requirements” (Pennsylvania State University Computer Integrated Construction Research Program: BIM Project Execution Planning Guide Version 2.0; July 2010, p. 99).

Such specifications are inadequate for ensuring the reliable exchange of the “information” – the object properties – embedded in the BIM. As the development of the experimental BIMs demonstrates, it is indispensable to examine the object types required and the property sets that must be associated with each type. Standardizing nomenclature is also important. Interoperability demands agreement not only on what object types are called but also what their properties are called.

BIM Execution Planning should therefore include development of a common object library to be utilized by all team members. Defining and standardizing the properties necessary for all activities and players is a substantial undertaking. The cost and time required to engage in this exercise for each project would be substantial. The more efficient approach is to build upon available open standards, such as IFC Model View Definitions (MVDs), COBie and CIS/2 to provide the framework. This approach is further facilitated if software vendors provide standards-

compliant templates. Alternatively, client organizations – principally owners with large capital programs – could create common object libraries for distribution to and use by project teams. Coupled with team training, this approach could get projects underway more quickly, support the efficient exchange of project information in computable form, and improve the quality and utility of the BIM data ultimately handed over to the client.

As the multiple property sets required for various building products become more widely understood and standardized, the industry should require manufacturers to provide this product information embedded in BIM objects available in open standard formats. The Specifiers Property Information Exchange (SPie) is a major initiative designed to define such standard property sets and encourage manufacturers to include them in their product models.

This model development plan was created to address specific project requirements, but it provides insights into a more general BIM execution plan framework. The following comments address the generalization of this model development approach.

Common Object Library: The common object library is developed to address the requirements of this project; specifically, to define consistent objects to be used in multiple building models, and to include common properties attached to the objects to produce the required deliverables in IFC and COBie formats. In a live design environment, there will be additional considerations, such as the different manufacturer-specific products and unique design alternatives available.

Additionally, this library was developed using a single software platform, Revit Architecture 2011 and Revit MEP 2011. While the objects can be exported to IFC format as a common data exchange format to be loaded into other design software, some of the object functionality specific to the software may not be available after the import. For example, Revit's native ability to parametrically control the width of a door opening is not available if the door was imported from an IFC file. Other software products may have similar limitations when working with imported elements.

COBie Parameters: It is necessary to add custom parameters to objects within Revit to store data required for a COBie format deliverable. This

will be required for most BIM authoring software, unless the vendor or a third party provides a COBie template. Each software platform includes options to extend the property sets in different ways. The project team should define which properties are required to be included in the model, as well as defining a process to exchange the BIMs without loss of data or functionality over the course of the project. For any project, there will be additional properties required for uses other than producing COBie deliverables: structural analysis, energy analysis, cost estimating, and so forth.

IFC Format: The Industry Foundation Classes (IFC) comprise an object oriented data model and a file format developed initially by the Industry Alliance for Interoperability and now by buildingSMART International (www.buildingsmart.com). The IFC model is registered by ISO as ISO/PAS 16739 and is in the process of becoming an official International Standard. It is used as a common data exchange format to facilitate interoperability between different software platforms covering the many disciplines that contribute to a building throughout its lifecycle. As an open format, IFC does not belong to a single software vendor; it is neutral and independent . It is a widely implemented format that can be read by most current BIM software products.

Despite its wide adoption, differences in the authoring software import/export routines can result in an IFC export or import that does not fully match the original source file or satisfy the intent of the data exchange. In order to be effective, every IFC exchange must follow an “exchange requirement” that specifies the information that needs to be present. In order to develop the common object library discussed in this chapter, exchange requirements were developed, based on the COBie deliverable specification.

When developing a BIM strategy using the IFC format for data exchange, it is important for the project team to test the proposed process and understand any software limitations.

1.9 Scope

The scope of this model development plan is to establish the process and guide the development of the required building models. It will also be a reference for future modeling projects.

2 Common Object Library Description

2.1 Objectives

In an active design practice, the development and maintenance of an object library is often an ongoing process. New objects must be added to meet the needs of different projects or to include additional functionality with each new release of the modeling software. For the purpose of this project, the common object library is developed to include objects needed to create the 3 required building models, along with the template files needed for each design discipline. The full list of objects and the process used to create and verify the library files is documented in this chapter.

2.2 Approach

It is important to define the relevant standards and requirements for the model files before the project team begins modeling. This includes modeling standards, file naming conventions, and any other aspect of the model that needs to be consistent across multiple files or specified for downstream use. Before a model object can be included in the common object library or used on a project, it must be reviewed and modified to comply with the modeling standards. There may also be additional considerations that are unique to the selected modeling tools and data exchanges that need to be addressed. The standards for this project are detailed below.

2.3 Scope

The scope of this common object library report is to define the requirements for and document the creation of a library of building components to be used to create a consistent set of building models based on a common modeling standard. It will also be a reference for future modeling projects.

2.4 Modeling Standards

Project file names – Building model files use the following naming format:

Table 2-1: Project File Name.

Project	Discipline	User Definable	Extension
PPP_	D_	U	EEE
Duplex_	A_	-	.rvt

Project – This field defines the building type:

Table 2-2: Project Name.

Code	Discipline
TEMPLATE	Common object library template file
DUPLEX	Duplex apartment building
OFFICE	Office building
CLINIC	Clinic building

Discipline – This field defines the design discipline of the model. The following discipline designations will be used:

Table 2-3: Discipline Abbreviations.

Code	Discipline
A	Architectural
S	Structural
M	Mechanical
E	Electrical
P	Plumbing
F	Fire Protection
MEP	MEP/FP Combined model

Common object library file names – Common object library component files will use the standard naming convention established by the object library that ships with the Revit 2011 software. The following naming format will be used:

Table 2-4: Family File Names.

Units	Description	Extension
U_	D	.EEE
M_	Single Flush	.rvt

Other standards and considerations – In a traditional cad environment, drawing layers are used to organize drawing objects and to control printed line weights, line styles, or line visibility within each cad file. Revit files do not use layers. Instead Revit includes global settings to control line weights, line styles, and object visibility based on the object category, or “Object Style”. This allows the user to define the presentation of all objects based on their category, rather than relying on the use of individual layers within each separate view or drawing file. While most traditional cad standards would include layer requirements, standards based on a Revit workflow may need to address Object Styles instead.

For the purpose of this project, all objects included in the common object library will use the default object style settings.

2.5 COBie Properties

In order to generate a COBie deliverable directly from a Revit model, custom object properties must be added to the model to carry additional COBie data that is not available in the default Revit configuration. When the Revit model is exported to IFC format, the parameters are exported as IfcPropertySingleValue, with the Ifc property name matching the Revit parameter name.

2.6 Family Parameters and Project Parameters

When adding parameters to objects in a Revit model, the new parameters may be applied to the individual families as a family parameter or to an entire object category as a project parameter. These custom parameters can be added to the objects using default Revit commands.

- Family parameters are applied to each individual component family by editing the family RFA file directly. If the parameter is required on multiple objects or object types, each individual file will need to be edited to include the new parameter. Likewise, if a new family is added to a project, it must be updated to include the custom parameters. For example, if a generic model element is replaced with a manufacturer provided family of a specific product, the new family will need to be updated to include the required parameters.
- Project parameters are applied to entire object categories within the Revit template. When a component file is loaded into the template or project file for use, project parameters are automatically added to the component based on the object category, so there is no need to add the parameters to each individual component file.

When adding parameters that apply to entire object categories, such as those required for basic COBie data, it is best to apply them as project parameters. This will create a set of common properties that are available on all objects of the desired category, without the need to update individual files. This will ensure consistency and data integrity of the individual family files are changed.

Family parameters can be used to extend the data set for unique objects within the broad object categories when a specific subset of properties is required. This includes object specific property fields required by the owner for FM use. For example, the owner may need to track a specific subset of properties for chillers and a different subset of properties for VAV boxes, both of which are included in the Revit object category for mechanical equipment.

For the purpose of this project, all of the baseline COBie properties have been added to the templates as project parameters.

2.7 Shared Parameters

When adding parameters to a Revit model, the new parameter can be added as a basic parameter or as a shared parameter. The designation of “shared parameter” is a method of standardizing the parameter format (naming standard and data type) across multiple files, and can apply to the parameters assigned in both the project and the family. When selecting the shared parameter option, the parameter name and settings can be defined in an external text file, allowing the same settings to be used across multiple files. In addition to ensuring consistency between files, the shared parameter option also allows the parameter to be used in tags when documenting a project.

All custom COBie data parameters have been added to the templates as shared parameters, with the exception of Category Code and Category Description, as described below. The resulting shared parameter text file *COBieSharedParameters.txt* is included with the common object library template files.

2.8 OmniClass Parameters

Revit component families include default fields for OmniClass Number and OmniClass Title, which can be set within the family file. However, Revit system families which are built into the template file (including Rooms, Walls, Roofs, etc), do not include fields for OmniClass information.

To address this, Category Code and Category Description project parameters have been added to the Rooms and Project Information categories in the Revit template files. These parameters are used to carry the OmniClass categories for spaces (in the Rooms category) and for the building (in the Project Information category). Additionally, shared parameters named Classification Code and Classification Description have been added to the system families to assign OmniClass data to the built in object types.

To facilitate applying the OmniClass category to the room objects, the template includes a Room Key Schedule for OmniClass Table 13. When editing the properties of a room, the OmniClass category can be selected from the *OmniClass Table 13 Category* property listed under the room’s Identity Data. This will automatically fill in the Category Code and

Category Description fields for the selected room. Note that these parameters are not added as shared parameters, because Revit 2011 does not allow access to shared parameters within a key schedule.

2.9 Instance Parameters and Type Parameters

Parameters can be added to Revit objects and categories as either instance parameters or type parameters. An instance parameter allows a unique input for every individual placement, or instance, of an object. A type parameter allows a single data input to be defined for all objects of the given type. All COBie properties from the Component tab, as well as any unique properties that are required on other tabs, are added as instance parameters. All COBie properties that are required on the Type tab, or are common to all objects of the same type, are added as type parameters. This setting is listed in the parameter groups defined in the *COBieSharedParameters.txt* file, and is listed in the charts in the next section.

2.10 IFC Export Override Parameters

When exporting a model to IFC format, Revit will map each object type from the model to the corresponding IFC entity type. The settings for these category mappings are edited in the IFC Export Classes dialog box, and can be saved as an external text file.

One limitation of this object category settings is that Revit object categories are defined more broadly than the corresponding IFC entities. For example, the Revit MEP category for Mechanical Equipment covers many types of equipment, including boilers, chillers, fans, etc. while the IFC entities include specific categories for IfcBoiler, IfcChiller, and IfcFan. When these objects are exported from Revit to IFC using the default settings, all of the objects are exported based on the single category defined in the IFC Export Classes dialog box.

In order to override an individual family's IFC export category, two additional shared project parameters have been added: IfcExportAs and IfcExportType.

- **IfcExportAs:** This parameter should be filled in with a valid IFC entity type. Revit will export the object to the IFC category given in this

- property instead of the default setting defined in the IFC Export Classes dialog box.
- **IfcExportType:** This parameter should be filled in with the IFC Predefined Type setting.

These parameters were added to the project templates as shared project parameters available to every object type. If these parameters are left empty, then the default export settings will be used. If they are filled in, then this setting will override the default IFC export category.

See section 0 of this chapter for a list of the families that used the IFC override settings.

2.11 Mapping of Revit Parameters to COBie Spreadsheet

The following charts list the COBie parameters used in the common object library template files. These parameters are exported to IFC, and are then mapped directly to a COBie spreadsheet using the BimServices Transfrom1 utility. The spreadsheet tabs that can be directly populated using the BimServices utility include: Facility, Floor, Space, Type, Component, and Attribute. Revit can also contain data required on the Zone and System tabs, but this data is not picked up by the BimServices utility or is not formatted in a way that will map directly to the tab layout. These tabs are addressed separately below. Data recorded on the remaining tabs are not generally available in a design-phase Revit model.

2.11.1 Facility Tab

Table 2-5: Facility Tab Parameters.

Facility Tab Parameters					
COBie Sheet / Field	Revit Property (Default)	Revit Property (Custom)	Rooms (Instance)	Project Information	Project Units
Name	See note 1				
CreatedBy	Revit Username				
CreatedOn	See note 2				
Category		Category Code	X	X	
Category		Category Description	X	X	
ProjectName	Project Number			X	
SiteName	Exported as “Default”				
LinearUnits	Project Units: Length				X
AreaUnits	See Note 2				X
VolumeUnits	See Note 2				X
CurrencyUnits	Not exported				X
AreaMeasurement	See note 3				
ExtSystem	IfcApplication				
ExtObject	By object type				
ExtIdentifier	Element Guid				
ExtSiteObject	IfcSite				
ExtSiteIdentifier	Guid				
ExtFacilityObject	IfcBuilding				
ExtFacilityIdentifier	Guid				
Description	Not exported				
ProjectDescription	Not exported				
SiteDescription	Exported as “Default”				
Phase	Project Status			X	

Notes:

1. The project name is available in the IFC file, but the field is not picked up by the BimServices Transform1 utility when generating the COBie file.
2. Length, area, volume and other units can be set in the Revit project file using the Project Units command, and these fields are populated by the BimServices Transform1 utility according to these settings. However, when Revit exports actual area and volume to IFC, it uses the Length unit setting as the default to calculate area and volume, regardless of the internal settings for area or volume. Length, area, and volume settings within Revit should be set to use the same base unit to maintain consistency.
3. Revit exports room area to IFC format using the project settings for Area and Volume Computations. By default, this is set to Wall Center, but it can also be set to Wall Finish, Wall Core Layer, or Wall Core Center. Revit does not list the area settings in the IFC export, but it lists GSA BIM Area as the area scheme.

2.11.2 Floor Tab

Table 2-6: Floor Tab Parameters.

Floor Tab Parameters			
COBie Sheet / Field	Revit Property (Default)	Revit Property (Custom)	Levels (Instance)
Name	Level Name		X
CreatedBy	Revit Username		
CreatedOn	See note 1		
Category		Category	X
ExtSystem	IfcApplication		
ExtObject	By object type		
ExtIdentifier	Element Guid		
Description	Level Name		
Elevation	Elevation		X
Height	N/A		

Notes:

1. This field is populated by the BimServices Transform1 utility when the COBie file is generated, and is not based on a Revit property.

2.11.3 Space Tab

Table 2-7: Space Tab Parameters.

Space Tab Parameters			
COBie Sheet / Field	Revit Property (Default)	Revit Property (Custom)	Rooms (Instance)
Name	Room Number		X
CreatedBy	Revit Username		
CreatedOn	See note 1		
Category		Category Code	X
Category		Category Description	X
FloorName	Level: Name		X
Description	Room Name		X
ExtSystem	IfcApplication		
ExtObject	By object type		
ExtIdentifier	Element Guid		
Room Tag	N/A		
UsableHeight	Unconnected Height		X
GrossArea	Area		X
NetArea	N/A		X

Notes:

1. This field is populated by the BimServices Transform1 utility when the COBie file is generated, and is not based on a Revit property.

2.11.4 Type Tab

Table 2-8: Type Tab Parameters.

Type Tab Parameters			
COBie Sheet / Field	Revit Property (Default)	Revit Property (Custom)	All Components (Type)*
Name	Family Type		X
CreatedBy	Revit Username		
CreatedOn	See note 1		
Category	OmniClass Number	See Note 2	X
Category	OmniClass Title	See note 2	X
Description	Family Type		X
AssetType		AssetAccountingType	X
Manufacturer	Manufacturer		X
ModelNumber		ModelNumber	X
WarrantyGuarantorParts		WarrantyGuarantorParts	X
WarrantyDurationParts		WarrantyDurationParts	X
WarrantyGuarantorLabor		WarrantyGuarantorLabor	X
WarrantyDurationLabor		WarrantyDurationLabor	X
ExtSystem	IfcApplication		
ExtObject	By object type		
ExtIdentifier	Element Guid		
ReplacementCost		ReplacementCost	X
ExpectedLife		ExpectedLife	X
DurationUnit		DurationUnit	X
WarrantyDescription		WarrantyDescription	X
NominalLength		NominalLength	X
NominalWidth		NominalWidth	X

Type Tab Parameters			
COBie Sheet / Field	Revit Property (Default)	Revit Property (Custom)	All Components (Type)*
NominalHeight		NominalHeight	X
ModelReference		ModelReference	X
Shape		Shape	X
Size		Size	X
Color		Color	X
Finish		Finish	X
Grade		Grade	X
Material		Material	X
Constituents		Constituents	X
Features		Features	X
AccessibilityPerformance		AccessibilityPerformance	X
CodePerformance		CodePerformance	X
SustainabilityPerformance		SustainabilityPerformance	X

Notes:

1. This field is populated by the BimServices Transform1 utility when the COBie file is generated, and is not based on a Revit property.
2. OmniClass Number and OmniClass Title are default parameters available in Revit component families, but are not available on all system families.

2.11.5 Component Tab

Table 2-9: Component Tab Parameters.

Component Tab Parameters			
COBie Field	Revit Property (Default)	Revit Property (Custom)	All Components (Instance)
Name	See note 1		X
CreatedBy	Revit Username		
CreatedOn	See note 2		
TypeName	Family Type		X
Space	Room: Name		X
Description	See note 1		
ExtSystem	IfcApplication		
ExtObject	By object type		
ExtIdentifier	Element Guid		
SerialNumber		SerialNumber	X
InstallationDate		InstallationDate	X
WarrantyStartDate		WarrantyStartDate	X
TagNumber		TagNumber	X
BarCode		BarCode	X
AssetIdentifier		AssetIdentifier	X

Notes:

1. Element Name is exported to IFC as a compound field composed of the following Revit fields: Family: Family Type: Element ID
2. This field is populated by the BimServices Transform1 utility when the COBie file is generated, and is not based on a Revit property.

2.11.6 Attribute Tab

Any additional parameters from Revit families are included on the Attribute tab. The available parameters will vary by Revit family and object type, but the following properties are included at a minimum to address COBie Attribute tab fields for each object type.

Table 2-10: Attribute Tab Parameters.

Attribute Tab Parameters		All Components (Type)	Doors (Type)	Windows (Type)	Rooms (Instance)	Furniture (Type)
Revit Property (Default)	Revit Property (Custom)					
	Reference	X				
	ArticleNumber	X				
	ProductionYear	X				
Construction Type			X	X		
	OperationType		X	X		
	Area		X	X		X
Fire Rating			X			
	GlazingAreaFraction		X	X		
	IsFireExit		X			
	ConfigurationType			X		
	FireRating			X		
	FloorCovering				X	
	CeilingCovering				X	
	WallCovering				X	
	LoadCapacity				X	

2.11.7 Zone Tab

The Zone tab in the COBie spreadsheet is used to group Spaces together based on common function, use, or other requirements. Default Zone types include Circulation Zone, Lighting Zone, Fire Alarm Zone, Historical Preservation Zone, Occupancy Zone, and Ventilation Zone, which are defined on the PickList tab of the COBie spreadsheet. Revit MEP includes an HVAC Zone tool which allows the MEP designer to group spaces together for the purpose of designing the mechanical systems. This information would correspond to the Ventilation Zone type on the COBie Zone tab. However, Revit does not export HVAC Zone data to IFC, and BimServices does not populate this tab from the IFC model.

To address the requirements of the Zone tab, a series of project parameters have been added to Rooms in the Revit template files to assign each room to each zone type. These parameters will export to IFC as IfcPropertySingleValue, and will be picked up by the BimServices Transfrom1 utility on the Attribute tab. These properties can be used as a reference to manually populate the Zone tab. If additional zones are required on a project, additional properties must be added to the Rooms category using the Project Parameters command.

Table 2-11: Zone Tab Parameters.

Zone Tab Parameters		
COBie Zone Type	Revit Property (Custom)	Rooms (Instance)
Circulation Zone	CirculationZoneName	X
Lighting Zone	LightingZoneName	X
Fire Alarm Zone	FireAlarmZoneName	X
Historical Preservation Zone	HistoricalPreservationZoneName	X
Occupancy Zone	OccupancyZoneName	X
Ventilation Zone	VentilationZoneName	X

2.11.8 System Tab

The System tab in the COBie spreadsheet is used to group components into building systems. Revit MEP includes systems for mechanical, piping, and electrical elements, allowing the user to create multiple systems within each category.

When exporting a model to IFC, Revit will include properties on the MEP elements for System Type and System Name, which list the system(s) that each item is associated with. These properties will be picked up by the BimServices Transform1 utility on the Attribute tab. These properties can be used as a reference to manually populate the System tab.

2.12 Common Object Library Files

The common object library files were selected from free publicly available sources or created specifically for use on this project. Model objects were collected or created for use in Revit Architecture 2011 and Revit MEP 2011, from the following sources in order of preference:

- **Revit 2011 Content Library:** Most objects can be taken directly from the default content library that is installed with Revit 2011 products.
- **Free online libraries:** Objects not available in the default library can often be found online at websites dedicated to the model authoring tools. In this case, the primary online source was www.seek.autodesk.com, a content search engine that allows users to search for content from multiple online sources for use in AutoDesk software such as Revit.
- **Custom creation:** If the required objects were not available in a free public library, they were created in Revit Architecture 2011 or Revit MEP 2011 for use on this project.
- Model content within Revit can be grouped into two broad categories:
- **Component families** are elements that are placed individually and can be created and saved as external files. They include building objects such as doors, windows, furniture, etc. Component families are saved as individual files in the Revit .rfa file format, and are provided as individual files in the common object library.
- **System families** are elements that are defined within the Revit interface to serve a specific purpose as part of the building such as walls, floors, and roofs. System families are built into the Revit interface and cannot be saved as separate files. These families are

included in the discipline templates provided in the Revit .rte file format.

The common object library families and the source for each are given in the tables below.

2.13 Revit Component Families

Table 2-12: Component Family File Names.

Revit Component Families: Air Terminals		
Family File Name	Source	Comments
M_Louver - Extruded.rfa	MEP Library	
M_Return Diffuser.rfa	MEP Library	
M_Supply Diffuser – Sidewall.rfa	MEP Library	
M_Supply Diffuser.rfa	MEP Library	
Revit Component Families: Casework		
Family File Name	Source	Comments
M_Base Cabinet-Double Door & 2 Drawer.rfa	Arch Library	
M_Counter Top w Sink Hole.rfa	Arch Library	
M_Counter Top.rfa	Arch Library	
M_Counter Top-L Shaped.rfa	Arch Library	
M_Tall Cabinet-Single Door(2).rfa	Arch Library	
M_Upper Cabinet-Double Door-Wall.rfa	Arch Library	
M_Upper Cabinet-Single Door-Wall.rfa	Arch Library	
M_Vanity Cabinet-Double Door Sink Unit.rfa	Arch Library	
M_Vanity Counter Top w Round Sink Hole.rfa	Arch Library	
Revit Component Families: Data Devices		
Family File Name	Source	Comments
M_Data Outlet.rfa	MEP Library	
M_Ethernet Switch.rfa	MEP Library	
Revit Component Families: Doors		
Family File Name	Source	Comments
M_Curtain Wall Dbl Chain Link.rfa	Custom	
M_Curtain Wall Dbl Glass.rfa	Arch Library	
M_Curtain Wall Sgl Glass.rfa	Arch Library	

M_Double-Flush.rfa	Arch Library	
M_Double-Glass 1.rfa	Arch Library	
M_Roof Hatch.rfa	Autodesk Seek	
M_Single-Flush.rfa	Arch Library	
M_Single-Glass 1.rfa	Arch Library	
M_Toilet Partition.rfa	Custom	

Revit Component Families: Electrical Equipment

Family File Name	Source	Comments
M_Diesel Emergency Power Generator.rfa	MEP Library	
M_Lighting and Appliance Panelboard - 208V MLO.rfa	MEP Library	
M_Transformer Switchboard.rfa	MEP Library	

Revit Component Families: Electrical Fixtures

Family File Name	Source	Comments
M_Elevator Door-Center.rfa	Autodesk Seek	
M_Elevator-Hydraulic.rfa	Autodesk Seek	
M_Junction Boxes - Load.rfa	MEP Library	
M_Duplex Receptacle.rfa	MEP Library	
M_Microwave.rfa	MEP Library	
M_Range.rfa	Arch Library	
M_Refrigerator.rfa	MEP Library	
M_Lighting Switches.rfa	MEP Library	
M_Thermostat.rfa	MEP Library	

Revit Component Families: Fire Alarm Devices

Family File Name	Source	Comments
M_Fire Alarm Control Panel.rfa	MEP Library	
M_Fire Alarm Strobe Speaker - Wall Mounted.rfa	MEP Library	
M_Manual Pull Station.rfa	MEP Library	
M_Smoke Detector.rfa	MEP Library	

Revit Component Families: Furniture

Family File Name	Source	Comments
M_Bed-Standard.rfa	Arch Library	
M_Chair-Corbu.rfa	Arch Library	
M_Marker Board.rfa	Autodesk Seek	

M_Shelving.rfa	Arch Library	
M_Sofa.rfa	Arch Library	
M_Table-Coffee.rfa	Arch Library	

Revit Component Families: Lighting Fixtures

Family File Name	Source	Comments
M_Ceiling Light - Flat Round.rfa	MEP Library	
M_Downlight - Recessed Can.rfa	MEP Library	
M_Exit Sign.rfa	MEP Library	
M_Pendant Light - Hemisphere.rfa	MEP Library	
M_Pendant Light - Linear - 2 Lamp.rfa	MEP Library	
M_Plain Recessed Lighting Fixture.rfa	MEP Library	
M_Sconce Light - Sphere.rfa	MEP Library	
M_Troffer Light - Lens.rfa	MEP Library	

Revit Component Families: Mechanical Equipment

Family File Name	Source	Comments
M_Air Handling Unit - Split System - Horizontal.rfa	MEP Library	
M_Air Handling Unit-Vertical Packaged-DX-21-35 kW.rfa	MEP Library	
M_Centrifugal Fan - Rooftop - Upblast.rfa	MEP Library	
M_Hot Water Boiler - 59-440 kW.rfa	MEP Library	
M_Inline Pump - Circulator.rfa	MEP Library	
M_Radiator - Hydronic Fin Tube.rfa	MEP Library	
M_Screw Chiller - Air Cooled - 1406-1758 kW.rfa	MEP Library	
M_Sewage Pump - Vertical Discharge.rfa	MEP Library	
M_VAV Unit - Single Duct.rfa	MEP Library	
M_Water Heater.rfa	MEP Library	

Revit Component Families: Plumbing Fixtures

Family File Name	Source	Comments
M_ADA shower Seat.rfa	Autodesk Seek	
M_Backflow Preventer - 15-50 mm.rfa	MEP Library	
M_Ball Valve - 50-150 mm.rfa	MEP Library	
M_Bath Tub.rfa	MEP Library	
M_Cleanout Two-Way - PVC - Sch 40 - DWV.rfa	MEP Library	
M_Drinking Fountain - Rectangular - Wall Mounted.rfa	MEP Library	

M_Fire Extinguisher Cabinet.rfa	MEP Library	
M_Floor Drain - Round.rfa	MEP Library	
M_Grab Bars.rfa	Autodesk Seek	
M_Hand Dryer.rfa	MEP Library	
M_Kitchen Unit.rfa	MEP Library	
M_Lavatory - Oval.rfa	MEP Library	
M_Mirror.rfa	Autodesk Seek	
M_Roof Drain.rfa	MEP Library	
M_Shower Stall - Rectangular.rfa	MEP Library	
M_Sink - Island - Single.rfa	MEP Library	
M_Sink - Kitchen - Double.rfa	MEP Library	
M_Sink - Work.rfa	MEP Library	
M_Soap Dispenser.rfa	Autodesk Seek	
M_Toilet Paper Holder.rfa	Autodesk Seek	
M_Towel Dispensers - receptical.rfa	Autodesk Seek	
M_Towel Dispensers.rfa	Autodesk Seek	
M_Urinal - Wall Hung.rfa	MEP Library	
M_Water Closet - Flush Tank.rfa	MEP Library	
M_Water Closet - Flush Valve - Wall Mounted.rfa	MEP Library	

Revit Component Families: Sprinklers

Family File Name	Source	Comments
M_Sprinkler - Horizontal Sidewall.rfa	MEP Library	
M_Sprinkler - Pendent.rfa	MEP Library	

Revit Component Families: Structural Columns

Family File Name	Source	Comments
M_Concrete-Rectangular-Column.rfa	Arch Library	
M_W-Wide Flange-Column.rfa	Arch Library	See note 1

Revit Component Families: Structural Foundations

Family File Name	Source	Comments
M_Footing-Rectangular.rfa	Arch Library	

Revit Component Families: Structural Framing

Family File Name	Source	Comments
M_K-Series Bar Joist-Rod Web.rfa	Arch Library	See note 1

M_Plywood Web Joist.rfa	Arch Library	
M_W-Wide Flange.rfa	Arch Library	See note 1
Revit Component Families: Telephone Devices		
Family File Name	Source	Comments
M_Telephone Outlet.rfa	MEP Library	
M_Telephone Terminal Board.rfa	MEP Library	
Revit Component Families: Windows		
Family File Name	Source	Comments
M_Casement.rfa	Custom	
M_Fixed.rfa	Arch Library	
M_Skylight.rfa	Arch Library	

Notes:

1. This object type is available in a wide range of sizes. Rather than create all of the different types in the main RFA file, Revit allows the individual types to be defined in an external TXT file called a *type catalog*, to allow the user to select the desired sizes without loading every possible type. The type catalog must be saved in the same directory as the associated RFA file and have the same file name. The type catalog file is included in the common object library.

2.14 Revit System Families

System families are contained within the Revit templates. Many of these objects are not treated as assets by the BimServices Transform1 utility, but are included in the common object library as individual IFC exports. The system families used and their host templates are listed below.

Table 2-13: System Family Names.

Revit System Families: Cable Trays		
Family File Name	Source	Comment
Cable Tray with Fittings Wire mesh Cable Tray	TEMPLATE_E.rte	
Revit System Families: Ceilings		
Family File Name	Source	Comment
Compound Ceiling ACT 600 x 600mm Grid	TEMPLATE_A.rte	
Compound Ceiling Gypsum Board	TEMPLATE_A.rte	

Revit System Families: Conduits		
Family File Name	Source	Comment
Data	TEMPLATE_E.rte	
Electrical Metallic Tubing (EMT)	TEMPLATE_E.rte	
Revit System Families: Ducts		
Family File Name	Source	Comment
Rectangular Duct Mitered Elbows - Taps	TEMPLATE_M.rte	
Rectangular Duct Mitered Elbows - Tees	TEMPLATE_M.rte	
Round Duct - Taps	TEMPLATE_M.rte	
Round Duct - Tees	TEMPLATE_M.rte	
Revit System Families: Flex Ducts		
Family File Name	Source	Comment
Flex - Round.ifc	TEMPLATE_M.rte	
Revit System Families: Floors		
Family File Name	Source	Comment
66mm Concrete With 38mm Metal Deck	TEMPLATE_A.rte	
127mm Slab on Grade	TEMPLATE_A.rte	
150mm Exterior Slab on Grade	TEMPLATE_A.rte	
150mm Slab on Grade	TEMPLATE_A.rte	
Finish Floor - Ceramic Tile	TEMPLATE_A.rte	
Finish Floor - Slate Tile	TEMPLATE_A.rte	
Finish Floor - VCT	TEMPLATE_A.rte	
Finish Floor - Wood	TEMPLATE_A.rte	
Residential - Wood Joist with Subflooring	TEMPLATE_A.rte	
Revit System Families: Pipes		
Family File Name	Source	Comment
Cold Water	TEMPLATE_P.rte	
Fire Protection	TEMPLATE_P.rte	
Hot Water	TEMPLATE_P.rte	
Storm	TEMPLATE_P.rte	
Vent	TEMPLATE_P.rte	
Waste	TEMPLATE_P.rte	

Revit System Families: Railings		
Family File Name	Source	Comment
900mm Handrail Only	TEMPLATE_A.rte	
900mm Pipe Guard Rail	TEMPLATE_A.rte	
1100mm Guard Rail	TEMPLATE_A.rte	
Revit System Families: Ramps		
Family File Name	Source	Comment
150mm Concrete ADA Ramp	TEMPLATE_A.rte	
Revit System Families: Roofs		
Family File Name	Source	Comment
EPDM Membrane on Rigid Insul on Metal Deck	TEMPLATE_A.rte	
Exterior Canopy	TEMPLATE_A.rte	
Live Roof over Wood Joist Flat Roof	TEMPLATE_A.rte	
Standing Seam Metal Roof	TEMPLATE_A.rte	
Revit System Families: Stairs		
Family File Name	Source	Comment
Concrete Pan - 180mm Max Riser 280mm Tread	TEMPLATE_A.rte	
Monolithic Concrete Stair	TEMPLATE_A.rte	
Residential - 200mm Max Riser 250mm Tread	TEMPLATE_A.rte	
Revit System Families: Structural Foundations		
Family File Name	Source	Comment
Bearing Footing - 900 x 300	TEMPLATE_A.rte	
Retaining Footing - 600 x 300 x 300	TEMPLATE_A.rte	
Revit System Families: Walls		
Family File Name	Source	Comment
Exterior - Brick on Brick	TEMPLATE_A.rte	
Exterior - Brick on Mtl Stud	TEMPLATE_A.rte	
Exterior - Insul Panel on Mtl Stud	TEMPLATE_A.rte	
Foundation - Concrete (264mm)	TEMPLATE_A.rte	
Foundation - Concrete (300mm)	TEMPLATE_A.rte	
Foundation - Concrete (350mm)	TEMPLATE_A.rte	
Foundation - Concrete (417mm)	TEMPLATE_A.rte	
Foundation - Concrete (550mm)	TEMPLATE_A.rte	

Interior - CMU Rated 2-HR	TEMPLATE_A.rte	
Interior - Furring (38 mm Stud)	TEMPLATE_A.rte	
Interior - Furring (152 mm Stud)	TEMPLATE_A.rte	
interior - Partition (92mm Stud)	TEMPLATE_A.rte	
Interior - Plumbing (152mm Stud)	TEMPLATE_A.rte	
Interior - Rated 1-HR (92mm Stud)	TEMPLATE_A.rte	
Interior - Toilet Partition (25mm)	TEMPLATE_A.rte	
Party Wall - CMU Residential Unit Demising Wall	TEMPLATE_A.rte	
Retaining - Concrete (300mm)	TEMPLATE_A.rte	
Storefront	TEMPLATE_A.rte	Curtain Wall

2.15 Revit Template Files

Table 2-14: Revit Template File Names.

Revit Template Files		
Template File Name	Discipline	Comments
TEMPLATE_A.rte	Architectural	Includes Structural
TEMPLATE_M.rte	Mechanical	
TEMPLATE_E.rte	Electrical	
TEMPLATE_P.rte	Plumbing	Includes Fire Protection
TEMPLATE_MEPRTE	MEP	Combined template
COBieSharedParameters.txt		Parameters file
IFC-exportlayers.txt		IFC Export settings

2.16 IFC File Export

Common object library files are provided in their native Revit format, along with an IFC export of each family and the COBie spreadsheet created from each IFC file.

To create each individual IFC file, each Revit family is placed into an empty Revit project. If the family is a hosted object, such as a door that is hosted by a wall, then the required host object must be drawn first, with the family placed on the host element.

In order for each of the custom data parameters to export to IFC, each parameter must be filled in. A custom API routine was written to fill each parameter with text reporting the parameter name. In a live project, this data would be filled in with the appropriate COBie data instead.

IFC files of each family were exported from Revit using the Revit Application Menu > Export > IFC command.

2.17 IFC Export Settings

When exporting to IFC, Revit includes settings to map object categories to IFC entities. The default mappings are focused mainly on architectural elements, while many of the MEP elements are set to export as *IfcBuildingElementProxy* entities by default. These proxy entities do not include full object type information, and should be mapped to the correct IFC entity categories instead. The export settings are saved as an external TXT file, which is included in the common object library templates directory named *IFC-exportlayers.txt*.

2.18 IFC Export Override

In addition to the IFC export category mappings, some of the families required IFC export override parameters to map the Revit family to a specific IFC category, as described in section 3 of this chapter. The families that used the override settings, and the IFC categories that were used, are given in the following chart.

Table 2-15: IFC Export Override Parameter Settings.

IFC Export Override Parameter Settings		
Revit Family	IfcExportAs	IfcExportType
M_Air Handling Unit - Split System - Horizontal.rfa	IfcFanType	CENTRIFUGALAIRFOIL
M_Air Handling Unit - Vertical Packaged - DX - 21-35 kW.rfa	IfcFanType	CENTRIFUGALAIRFOIL
M_Bath Tub.rfa	IfcSanitaryTerminalType	BATH
M_Centrifugal Fan - Rooftop - Upblast.rfa	IfcFanType	CENTRIFUGALAIRFOIL
M_Drinking Fountain - Rectangular - Wall Mounted.rfa	IfcSanitaryTerminalType	SANITARYFOUNTAIN
M_Duplex Receptacle.rfa	IfcOutletType	POWEROUTLET
M_Fire Alarm Control Panel.rfa	IfcElectricDistributionPoint	CONTROLPANEL
M_Floor Drain - Round.rfa	IfcPipeFittingType	ENTRY
M_Hot Water Boiler - 59-440 kW.rfa	IfcBoilerType	WATER
M_Inline Pump - Circulator.rfa	IfcPumpType	CIRCULATOR
M_Lavatory - Oval.rfa	IfcSanitaryTerminalType	SINK
M_Lighting and Appliance Panelboard - 208V MLO.rfa	IfcElectricDistributionPoint	DISTRIBUTIONBOARD
M_Microwave.rfa	IfcElectricApplianceType	MICROWAVE
M_Radiator - Hydronic Fin Tube.rfa	IfcHeatExchangerType	SHELLANDTUBE
M_Range.rfa	IfcElectricApplianceType	ELECTRICCOOKER
M_Refrigerator.rfa	IfcElectricApplianceType	FRIDGE_FREEZER
M_Roof Drain.rfa	IfcPipeFittingType	ENTRY
M_Screw Chiller - Air Cooled - 1406-1758 kW.rfa	IfcChillerType	AIRCOOL
M_Screw Chiller - Air Cooled - 281-1231 kW.rfa	IfcChillerType	AIRCOOL
M_Shower Stall - Rectangular.rfa	IfcSanitaryTerminalType	SHOWER
M_Sink - Island - Single.rfa	IfcSanitaryTerminalType	SINK
M_Sink - Kitchen - Double.rfa	IfcSanitaryTerminalType	SINK
M_Sink - Work.rfa	IfcSanitaryTerminalType	SINK
M_Telephone Terminal Board.rfa	IfcElectricDistributionPoint	DISTRIBUTIONBOARD
M_Thermostat.rfa	IfcControllerType	TWOPOSITION

M_Transformer Switchboard.rfa	IfcTransformerType	VOLTAGE
M_Urinal - Wall Hung.rfa	IfcSanitaryTerminalType	URINAL
M_VAV Unit - Single Duct.rfa	IfcFanType	NOTDEFINED
M_Water Closet - Flush Tank.rfa	IfcSanitaryTerminalType	WCSEAT
M_Water Closet - Flush Valve - Wall Mounted.rfa	IfcSanitaryTerminalType	WCSEAT
M_Water Heater.rfa	IfcTankType	PREFORMED

2.19 COBie File Conversion

Once the IFC files have been exported from Revit, they are translated to COBie format using the BimServices Transform1 utility along with the *_asCOBIE2.xml.xsl* file (dated 04/06/2011). This tool will generate an IFCxml file and a COBie 2.40 spreadsheet from the IFC file.

Each COBie spreadsheet should contain one Component tab entry and one Type tab entry for the object from the IFC file. In the case of hosted elements (such as doors and windows), there is also be an entry for the host element. The BimServices utility also populates the Attribute tab with additional object data, along the Contact, Facility, and Level tab for each of the files.

Each of the COBie spreadsheets was manually checked to verify that the Type and Component tabs had been correctly filled in. While most of the files were correctly translated to the COBie spreadsheet, there were a few exceptions, as noted below.

Object category not exported from Revit: Using the default IFC export settings, two of the Revit categories did not export to IFC correctly. The default Revit setting used an invalid IFC entity category, which was then skipped during the export process. To correct this, the export settings were revised to use valid IFC categories, and re-exported. The Revit categories and relevant settings are listed below. The updated settings are included in the *IFC-exportlayers.txt* file in the common object library Templates directory.

Table 2-16: IFC Export Categories.

Revit Category	Default Setting	Revised Setting
Cable Trays	IFCCableTraySegment	IfcCableCarrierSegment
Cable Tray	IFCCableTrayFitting	IfcCableCarrierSegmentFitting
Conduits	IFCConduitSegment	IfcCableSegment
Conduit Fittings	IFCConduitFitting	IfcCableSegment

Component tab entry with no Type tab entry: Some object categories produced COBie spreadsheet files that included a Component tab entry with no corresponding Type tab entry. The categories included system family elements that are generally large assemblies or parts of the building such as walls, floors, and roofs. These objects are typically not required as COBie assets, and are ignored by the BimServices Transform1 utility unless a command line switch is used to include all objects in the IFC file is used (using the “all=yes” switch in the command line, as noted in the program’s documentation). The Revit categories and the IFC export settings are given below, along with a note indicating whether the category is treated as an asset by the BimServices utility. This project will use the “all=yes” switch to include these objects in the COBie files.

Table 2-17: Component Families with no COBie Type.

Revit Category	Export Category	BimServices Asset
Ceiling	IfcCovering	Yes
Floor	IfcSlab	No
Generic Model	IfcBuildingElementProxy	See below
Railing	IfcRailing	No
Ramp	IfcRamp	No
Roof	IfcRoof	No
Specialty Equipment	IfcBuildingElementProxy	See below
Stair	IfcStair	No
Structural Foundation	IfcFooting	No
Structural Framing	IfcBeam	No
Wall	IfcWall	No

The Revit object categories for Generic Models and Specialty Equipment are very broad categories that cover many different building products and components. For example, Specialty Equipment families can include site furnishings, conveying equipment, toilet specialties, and many other product categories that are typically included as COBie assets. Using the default IFC export settings, these categories will export as IfcBuildingElementProxy entities, a generic IFC entity category. When BimServices creates the COBie spreadsheet from the IFC file, IfcBuildingElementProxy objects are picked up on the Component tab, but not on the Type tab.

To avoid creating unreferenced entries on the Component tab, the Revit categories for Generic Models and Specialty Equipment are not used in the common object library. Any family selected from one of these categories must be modified by opening the file in Revit and using the Family Category and Parameters command to change the object category. Note that this will change the object category in the Revit project file, possibly affecting functionality such as schedules and graphic display. This will also change the IFC export category, but will not affect the OmniClass category setting. The common object library files that have been changed from their original category settings are listed below.

Table 2-18: Component Families with Modified Category Settings.

Revit Family	Default Category	Revised Category
M_ADA shower Seat.rfa	Specialty Equipment	Plumbing Fixtures
M_Range.rfa	Specialty Equipment	Electrical Fixtures
M_Elevator Door-Center.rfa	Specialty Equipment	Electrical Fixtures
M_Elevator-Hydraulic.rfa	Specialty Equipment	Electrical Fixtures
M_Exit Sign.rfa	Specialty Equipment	Lighting Fixtures
M_Fire Extinguisher Cabinet.rfa	Specialty Equipment	Plumbing Fixtures
M_Grab Bars.rfa	Specialty Equipment	Plumbing Fixtures
M_Hand Dryer,rfa	Specialty Equipment	Plumbing Fixtures
M_Marker Board.rfa	Specialty Equipment	Furniture
M_Microwave.rfa	Specialty Equipment	Electrical Fixtures
M_Mirror.rfa	Specialty Equipment	Plumbing Fixtures
M_Refrigerator.rfa	Specialty Equipment	Electrical Fixtures
M_Roof Hatch.rfa	Generic Models	Doors

M_Soap Dispenser.rfa	Specialty Equipment	Plumbing Fixtures
M_Toilet Paper Holder.rfa	Specialty Equipment	Plumbing Fixtures
M_Towel Dispensers-receptical.rfa	Specialty Equipment	Plumbing Fixtures
M_Towel Dispensers.rfa	Specialty Equipment	Plumbing Fixtures

Note that in some cases, the revised category may be less appropriate than the original Specialty Equipment category. The revised categories were selected as the most appropriate alternative, given the limitations on how the model objects are defined and used. For example, the *M_Hand Dryer.rfa* family might be considered an Electrical Fixture. However, in live projects this item is typically included with toilet specialties, and is listed above as a Plumbing Fixture so that it will schedule and display with other toilet specialties in the project.

Curtain wall: Curtain wall elements in Revit are a unique object type. The curtain wall element is a container object that hosts the associated curtain panels and mullions. Each of these three object types is exported to IFC as listed in the chart below.

Table 2-19: Curtain Wall IFC Categories.

Revit Category	Export Category	Ifc Subcategory
Curtain Wall	IfcCurtainWall	IfcCurtainWall
Curtain Panels	IfcCurtainWall	IfcPlate
Curtain Wall Mullions	IfcCurtainWall	IfcMember

Note that Curtain Panels and Curtain Wall Mullions are exported as IfcPlate and IfcMember subparts of the IfcCurtainWall container element defined in the Revit export settings.

When curtain wall objects are translated to COBie using the BimServices Transform1 utility, the resulting COBie file does not include proper type relationships for the curtain wall objects. The IfcCurtainWall instance is picked up on the Component tab, but there is no corresponding type on the Type tab. The IfcPlate and IfcMember types and component entries are translated correctly. A review of the original IFC file shows that the IfcPlate and IfcMember entities include both instances and types, but there is no type defined for the IfcCurtainWall entity.

It should be noted that the curtain wall element itself is primarily a container element. The curtain mullions and curtain panels define the geometry and attributes of the actual model elements or building assets. The curtain wall element in Revit is a modeling convention to simplify model creation. As such, it could be considered a software artifact and ignored without losing any relevant data in the resulting COBie file.

Duplicate property data: The COBie spreadsheets produced by the BimServices Transform1 utility include a number of fields that report the IFC property data multiple times. The affected COBie fields are listed in the table below.

Table 2-20: BimServices Duplicate Data Fields.

COBie Tab	COBie Field	Comments
Facility	Category	
Floor	Height	Lists a Boolean value (True/False) twice
Space	Category	
Space	UsableHeight	
Type	CodePerformance	Reports any property with "code" in the name
Component	InstallationDate	
Component	WarrantyStartDate	
Component	TagNumber	
Component	BarCode	
Component	AssetIdentifier	

2.20 COBie File Issues Review

After creating a COBie file from the IFC export, the COBie files were tested using the BimServices Transform1 utility to convert the COBie file back to IFC using the *_fromCOBIE2.ifcxml.xsl* file (dated 04/06/2011) and the *_Issues.xhtml.xsl* file (dated 01/06/2010). This process reviews the COBie file and creates an Issues report in XHTML file format. These results are included with the common object library files. With a few notable exceptions listed below, all of the tested files reported Compliance or Adequate Compliance in the issues report.

There were some object types that did not process correctly when running the *_Issues.xhtml.xsl* tool. These objects produced an error in the command line utility, and the Issues tool was aborted without completing the report. This resulted in the creation of an empty XHTML report.

This error occurred in any tested file that included one of the listed elements, including any families that are hosted by these elements or any tested building models that included these objects. The error occurred regardless of whether the elements were included in the COBie file using the “all=yes” command line switch. The common object library files for these objects are listed in the table below.

Table 2-21: Files not Processed by BimServices Issues Check.

Files not Processed by BimServices Issues Check		
IFC File Name	Category	Comments
EPDM Membrane on Rigid Insul on Metal Deck.ifc	Roofs	
Exterior Canopy.ifc	Roofs	
Live Roof over Wood Joist Flat Roof.ifc	Roofs	
Standing Seam Metal Roof.ifc	Roofs	
Residential - 200mm Max Riser 250mm Tread.ifc	Stairs	
Concrete Pan - 180mm Max Riser 280mm Tread.ifc	Stairs	
M_Skylight.ifc	Window	Roof hosted
M_Centrifugal Fan - Rooftop - Upblast.ifc	Mech. Equip.	Roof hosted
M_Roof Hatch.ifc	Doors	Roof hosted

Unless this issue can be resolved with the BimServices application developer, the IFC files for the building models will need to be created without the above object types. The objects can still be included in the Revit model for the sake of completing the building model, but they will be turned off prior to exporting the building to IFC.

2.21 Common Object Library IFC Files

The common object library is developed in Revit 2011 using native software elements. These objects were then exported to IFC, including both geometry and data properties. While the objects can be exported to IFC format as a common data exchange format to be loaded into other

design software, some of the object functionality specific to the software may not be available after the import. For example, Revit's native ability to parametrically control the width of a door opening is not available if the door was imported from an IFC file. Other software products may have similar limitations when working with imported elements.

3 Duplex Apartment Model Description

3.1 Objectives

This chapter describes the creation and testing of BIM models for the duplex apartment building.

3.2 Approach

The duplex apartment building was modeled in Revit Architecture 2011 and Revit MEP 2011, based on the government-provided IFC model for the duplex prototype. The models were exported to IFC format, and the BimServices Transform1 utility was used to create a COBie spreadsheet for each model. The BimServices utility was also used to test the models for compliance using the BimServices Issues tool.

3.3 Scope

The scope of this chapter is to document the creation and testing of BIM files for the duplex building type. It will also be a reference for future modeling projects.

3.4 Modeling Standards

Model files for the duplex apartment building include two linked files: one containing the Architectural and Structural elements, and one containing the MEP elements. Additionally, a reference file was created by importing the government-furnished IFC file. This reference file is not part of the Revit BIM file set, but it was linked in and used as a reference to start the new models.

The duplex apartment BIM consists of the following files:

Table 3-1: Duplex Apartment BIM Files .

File Name	Discipline
Duplex_A.rvt	Architectural/Structural
Duplex_MEП.rvt	MEP Combined
Duplex_A_IFC-Import.rvt	For reference only

3.5 File Origin

All BIM files for the Duplex Apartment use the northwest corner of the building as the file origin point. When linking models together, the models will align when the Import Positioning option is set to Auto – Origin to Origin in the Import/Link Revit dialog box.

3.6 File Setup

3.6.1 Reference Model

The layout for the duplex apartment building is based on a government-furnished model provided in IFC format. The IFC model included the Architectural layout, along with plumbing fixtures and similar equipment. The IFC model did not include full MEP system layout.

In order to use the IFC file as reference within Revit, the file was imported into a Revit Architecture project file. The imported building was then moved to establish the northwest corner as the origin point, and the file was saved as a new Revit project named *Duplex_A_IFC-Import.rvt*.

3.6.2 Architectural Model

The architectural and structural model was created in Revit Architecture using the *Template_A.rvt* Revit template file. The *Duplex_A_IFC-Import.rvt* reference file was linked into the new file, using the Origin to Origin import position setting, to be used as reference when starting the model. The reference file was later unloaded using the Manage Links command within Revit. The link is saved in the Revit project file, and it can be reloaded when needed to compare the new model to the original reference file.

The architectural/structural model was developed in Revit Architecture 2011 using the components included in the Common Object Library.

3.6.3 MEP Model

The MEP model was created in Revit MEP using the *Template_MEPM.rvt* Revit template file. The *Duplex_A.rvt* architectural model was linked into the new file, using the Origin to Origin import position setting, to be used as reference when creating the MEP model.

The MEP model was developed in Revit MEP using the components included in the Common Object Library.

The Revit MEP model uses both Rooms and MEP Spaces, which are very similar Revit objects that both export as IfcSpace entities, and are both mapped to the COBie Spaces tab by BimServices. In a typical Revit MEP project, Rooms are used to match the Architectural model room layout while Spaces are used for the MEP design, and contain additional MEP properties for analysis. While both Rooms and Spaces export to IFC as IfcSpace entities, only the Room object is used to establish containment relationships for objects in the model.

3.7 IFC File Export

The duplex apartment files are provided in their native Revit format, along with an IFC export of each file and the COBie spreadsheet created from each file. IFC files of each model were exported from Revit using the Revit Application Menu > Export > IFC command. The IFC export was created using IFC export settings defined in the *IFC-exportlayers.txt* file included in the common object library.

In order for each of the custom data parameters to export to IFC, each parameter must be filled in. A proprietary custom API routine was used to fill each model object's COBie parameters with text reporting the parameter name. In a live project, this data would be filled in with the appropriate COBie data instead.

In addition to the automated routine above, the COBie Zone properties applied to the Rooms were manually edited to group the rooms into Zones to divide them into separate living units (Unit A and Unit B of the duplex). These properties are be picked up by the BimServices utility on the COBie Zone tab.

3.8 COBie File Conversion

Once the IFC files have been exported from Revit, they are translated to COBie format using the BimServices Transform1 utility along with the *_asCOBIE2.xml.xsl* file. This tool will generate an IFCxml file and a COBie 2.40 spreadsheet from the IFC file. This project used the “all=yes” command line switch in the BimServices utility, which includes all

building objects in the COBie output as noted in the program's documentation.

Each of the COBie spreadsheets was manually checked to verify that the relevant tabs had been correctly filled in. While most of the model objects were correctly translated to the COBie spreadsheet, there were a few exceptions, as noted below.

Missing Data on the Contact Tab: The only Contact tab information available by default in the Revit IFC export is the user name of the person who exported the file, listed as an IfcPerson. BimServices uses this information to create a single entry on the Contact tab and creates an auto-generated email address. Since there is no additional user data in the Revit IFC file, the remaining fields on the Contact tab are left as 'n/a' and must be manually filled in.

Missing Data on the Facility Tab: The following fields are not available by default in the Revit IFC export: SiteName, CurrencyUnit, Description, and SiteDescription. These fields must be manually filled in.

Element Type not Exported to IFC: Some object categories within Revit export to IFC without a Type definition. The categories include Revit system family elements that are generally large assemblies or parts of the building such as walls, floors, and roofs. BimServices handles these object types in two different ways:

Create Type as IfcMaterial: BimServices will read the IFC data for certain object categories and create an IfcMaterial entry on the Type tab for these objects.

Create Component Entry without Type: A few object categories are loaded correctly onto the Component tab, but BimServices does not create a Type for these categories.

The Revit categories and IFC export settings are given below, along with a note indicating whether BimServices will create an IfcMaterial entry for the category or load the Component without creating the Type entry.

Table 3-2: Revit Families with no IFC Type.

Revit Category	Export Category	Created IfcMaterial
Ceiling	IfcCovering	Yes
Floor	IfcSlab	Yes
Railing	IfcRailing	No
Roof	IfcRoof, IfcSlab	Yes
Stair	IfcStair, IfcStairFlight, IfcMember	No
Structural Foundation	IfcFooting	No
Structural Framing	IfcBeam	No
Wall	IfcWall	Yes

Type Tab Entry as IfcMaterial: In addition to creating IfcMaterial entries on the Type tab for Ceiling, Floor, Roof, and Wall categories as listed above, BimServices will also load Type tab entries for IfcMaterial entities defined in the IFC file. Since these are materials used in the Revit model, rather than actual model objects, they do not have the same Type data available, nor do they have Component tab entries. These materials are used on the Assembly tab to create assemblies for the Ceiling, Floor, Roof, and Wall categories noted above.

Component Tab Space Field Missing: Some of the Component tab entries are missing data in the Space field. In most cases, this is because the object in the Revit model is not contained within a Room and therefore the IFC export does not include any containment relationship for these objects. Examples include piping that is run within a wall or foundation walls below grade, because they do not contact a Room in the Revit model.

Stair and Railing elements are also missing the Space field, even though these objects are contained within a Room in the Revit model. When these objects are exported from the Revit model to an IFC file, the export does not include the containment relationship.

Missing Attribute tab properties: The Attribute tab includes additional properties associated with the entities in the IFC file. However, some of the Revit object categories were not picked up on the Attribute tab. The Attribute tab does not include any properties for the object categories that do not have a Type defined in the Revit IFC export file

(Ceiling, Floor, Railing, Roof, Stair, Structural Foundation, Structural Framing, and Wall categories).

3.9 COBie File Issues Review

After creating a COBie file from the IFC export, the COBie files were tested using the BimServices Transform1 utility to convert the COBie file back to IFC using the *_fromCOBIE2.ifcxml.xsl* file and the *_Issues.xhtml.xsl* file. This process reviews the COBie file and creates an Issues report in XHTML file format. These results are included with the duplex model files.

When running the *_fromCOBIE2.ifcxml.xsl* file to create an IFC file to test, the IfcRoof entity creates an error that results in an invalid IFC file. Since the *_fromCOBIE2.ifcxml.xsl* file will create both an IFC file and an IFCXML file, the IFCXML file was used to produce the Issues report.

The results of the Issues report show that the COBie files produced Compliance or Adequate Compliance, with the exception of the missing fields and Type information noted in section 3.8 of this chapter.

3.10 Manual Review of COBie Files

In addition to the BimServices review, the COBie files produced by the BimServices utility were manually reviewed against the objects in the Revit model. To facilitate this review, multi-category schedules were created in Revit to report the objects and data that are exported to IFC and used by BimServices to create the COBie files. Multi-category schedules allow Revit to include objects from multiple categories in a single schedule. This can simplify comparisons to the COBie spreadsheets.

The schedules are included in the native Revit model files, as well as in the common object library templates as Excel files. Note that the schedules produced within Revit are not exact duplicates of the spreadsheets produced by the BimServices utility due to some formatting changes in the file conversion process, but they list the same properties read by the BimServices utility when creating the COBie spreadsheets.

In order to compare the Revit schedules to the COBie spreadsheets, each schedule was exported from Revit and opened in Microsoft Excel. The resulting Excel files are included with the duplex apartment files.

3.11 Architectural Model Review

The Architectural COBie file *Duplex_A_20110525_asCOBIE2.xls* produced by BimServices was compared to the multi-category schedules produced in the *Duplex_A.rvt* Revit file and their corresponding Excel file exports. Due to limitations of the Revit multi-category schedule, which does not include some system family categories like Floors, Roofs, and Ceilings, the COBie spreadsheet included many objects that are not listed in the Revit schedules.

Comparison of the objects and data available in the Revit schedule to the spreadsheet data in the COBie file shows that the data translation through IFC is consistent, with the exception of the missing Type entries and missing Attribute entries noted in section 3.8 of this chapter.

Direct comparison to the Revit file object quantities shows that there are 211 objects in the Revit Architecture file, while the COBie spreadsheet lists 216 objects on the Component sheet. Further inspection of the COBie file shows that Stair objects produce multiple entries in the COBie spreadsheet, while they are treated as a single object in Revit. The Stair object is exported to IFC with separate entities for the IfcStair (overall stair object), IfcStairFlight (individual flights), and IfcMember (stringers). This results in separate Component entities for each of these objects, while the Revit file contains just a single Stair object.

With the different structure of the Stair objects taken into account, the number of entities listed on the COBie Component spreadsheet matches the number of entities in the original Revit model.

Review of the COBie Type tab shows that most Components are properly related to a Type tab entry, with the exception of the categories noted in section 3.8 of this chapter. This can also be seen on the Component tab, where the TypeName field for these objects is listed as “n/a”. These object types would need to be manually created to complete the COBie Type tab, or BimServices would need to create Type entries from the component data.

3.12 MEP Model Review

The MEP COBie file *Duplex_MEPC_20110525_asCOBIE2.xls* produced by BimServices was compared to the multi-category schedules produced in

the *Duplex_MEП.rvt* Revit file and their corresponding Excel file exports. Elements in the MEP object categories are not subject to the limitations of the multi-category schedules found in the architectural file; all of the MEP objects are included in the multi-category schedules.

There are 926 model objects in the Revit MEP model, and there are also 926 entities listed on the COBie spreadsheet Component tab produced by BimServices. The Component list includes all of the objects used in the original Revit model.

Comparison of the objects and data available in the Revit schedule to the spreadsheet data in the COBie file shows that the data translation through IFC is consistent.

Review of the COBie Type tab shows that the MEP Components are all properly related to a Type tab entry. However, many of the Revit MEP system families are exported to IFC format with a unique Type entry created for every Component. This would need to be manually corrected in the COBie file.

The affected Revit categories are listed in the table below.

Table 3-3: MEP System Families with Multiple COBie Types.

Revit Category	Export Category
Conduits	IfcCableSegmentType
Pipes	IfcPipeSegmentType
Ducts	IfcDuctSegmentType

4 Clinic Model Description

4.1 Objectives

This chapter describes the creation and testing of BIM models for the clinic building.

4.2 Approach

The clinic building was modeled in Revit Architecture 2011 and Revit MEP 2011, based on the government-provided IFC model and 2D DWG files for the clinic prototype. The models were exported to IFC format, and the BimServices Transform1 utility was used to create a COBie spreadsheet for each model. The BimServices utility was also used to test the models for compliance using the BimServices Issues tool.

4.3 Scope

The scope of this chapter is to document the creation and testing of BIM files for the clinic building type. It will also be a reference for future modeling projects.

4.4 Modeling Standards

Model files for the clinic building include three linked files: one contains the Architectural elements, one contains the Structural elements, and one contains the MEP/FP elements. Additionally, a reference file was created by importing the government-furnished IFC file. This reference file is not part of the Revit BIM file set, but it was linked in and used as a reference to start the new models.

The government-provided reference files for the clinic building also included 2D DWG files that were used as reference when creating the Revit models. Within each discipline model, the DWG file for each plan drawing was linked into a Revit plan view. These linked files were used as backgrounds when creating the model objects within Revit. Once each Revit model was completed, the DWG files were removed to reduce the size of the Revit file and improve performance.

The clinic BIM consists of the following files:

Table 4-1: Clinic BIM Files .

File Name	Discipline
Clinic_A.rvt	Architectural
Clinic_S.rvt	Structural
Clinic_MEП.rvt	MEP/F Combined
Clinic_A_IFC-Import.rvt	For reference only

In addition to the model files listed above, it was necessary to divide the MEP model into smaller files. When processing the IFC files using the BimServices Transform1 utility, any IFC file larger than approximately 25 MB (depending on the model content) caused an “Out of Memory” error and failed to run correctly. In order to create IFC files of less than 25 MB, the MEP models were divided into partitions based on discipline and building area. This process is described in section 4.8 of this chapter.

The MEP models were divided into the following partitions:

Table 4-2: Clinic MEP Model Partitions .

File Name	Description
Clinic_M_FL01North.rvt	Mechanical, first floor north
Clinic_M_FL01South.rvt	Mechanical, first floor south
Clinic_M_FL02North.rvt	Mechanical, second floor north
Clinic_M_FL02South.rvt	Mechanical, second floor south
Clinic_E.rvt	Electrical (full building)
Clinic_P_UnderSlabNorth.rvt	Underground plumbing, north
Clinic_P_UnderSlabSouth.rvt	Underground plumbing, south
Clinic_P_FL01North.rvt	Plumbing, first floor north
Clinic_P_FL01South.rvt	Plumbing, first floor south
Clinic_P_FL02.rvt	Plumbing, second floor
Clinic_F.rvt	Fire protection (full building)

Note that these partition files are not intended to replace the *Clinic_MEП.rvt* file from the main BIM file list. Some Revit functionality

is lost when the building systems are created in separate files, as described in section 4.8 of this chapter.

4.5 File Origin

All BIM files for the clinic use the intersection of column grids A/1 as the file origin point. When linking models together, the models will align when the Import Positioning option is set to Auto – Origin to Origin in the Import/Link Revit dialog box.

4.6 File Setup

4.6.1 Reference Model

The layout for the clinic building is based on a government-furnished model provided in IFC format, along with government-furnished 2D Autocad DWG files. The 2D DWG files included architectural, mechanical, plumbing, and electrical layouts, and were used as the primary reference for developing the models.

In order to use the 2D DWG files within Revit, they were brought in to the active models using the Link command. Since the DWG files used different origin points, the files were then manually moved to align with the structural grid in the Revit file. Linking or inserting DWG files into a Revit model can impact the size and performance of the file. Once the Revit model was complete, the DWG files were removed to reduce the size of the Revit file and improve performance.

In order to use the IFC file as reference within Revit, the file was imported into a Revit Architecture project file. The imported building was then moved to establish the intersection of column grid A/1 as the origin point, and the file was saved as a new Revit project named *Clinic_A_IFC-Import.rvt*.

4.6.2 Structural Model

The structural model was created in Revit Architecture using the *Template_A.rvt* Revit template file. The DWG files for the structural plans were linked in to the Revit file and moved to align column grid A/1 with the Revit file origin point. These linked files were used as a reference to create the structural model, including column grids, foundations, framing, and slabs, and were removed from the file after the model was completed.

After the architectural and MEP models were developed, they were linked into the structural model for coordination. Using the elements in the architectural and MEP models as reference, the structural elements were updated to coordinate across the files. For example, the openings and edges of the floor and roof slabs were updated to match the stair and shaft openings in the architectural model.

The structural model was developed in Revit Architecture 2011 using the components included in the Common Object Library. Note that in a live project, the structural model would be created in Revit Structure, which includes additional tools for analysis and documentation of structural elements.

4.6.3 Architectural Model

The architectural model was created in Revit Architecture using the *Template_A.rvt* Revit template file. The DWG files for the architectural plans were linked in to the Revit file and moved to align column grid A/1 with the Revit file origin point. These linked files were used as a reference to create the architectural model, and were removed from the file after the model was completed.

In addition to linking the cad plans into the architectural model file, the *Clinic_S.rvt* Revit model was also linked into the architectural file, using the Origin to Origin import position setting, to provide the structural elements. Using Revit's Copy/Monitor tool, the grids and levels from the structural model were copied into the architectural model. This ensures that the datum elements (grids and levels) remain consistent across the files, reducing the need for manual adjustment and eliminating potential inconsistencies.

After the MEP model was developed, it was linked into the architectural model for coordination. Using the elements in the MEP model as reference, the architectural elements were updated to coordinate across the files. For example, the furring walls at downspouts and shafts were adjusted to match the MEP layout.

To verify the new model against the government-provided IFC model, the *Clinic_A_IFC-Import.rvt* reference file was linked into the new file using the Origin to Origin import position setting. The new model was reviewed against the government-furnished model to verify that the new model is

consistent with the original. The linked file was used as a reference to create the architectural model, and was removed from the file after the model was completed.

The architectural model was developed in Revit Architecture 2011 using the components included in the Common Object Library.

4.6.4 MEP Model

The MEP model was created in Revit MEP using the *Template_MEП.rvt* Revit template file. The *Clinic_A.rvt* and the *Clinic_S.rvt* architectural and structural models were linked into the new file, using the Origin to Origin import position setting, to be used as reference when creating the MEP model. Using Revit's Copy/Monitor tool, the grids and levels from the structural model were copied into the architectural model.

The DWG files for the MEP plans were linked in to the Revit MEP file and moved to align column grid A/1 with the Revit file origin point. These linked files were used as a reference to create the MEP model, and were removed from the file after the model was completed.

The MEP model was developed in Revit MEP using the components included in the Common Object Library.

4.7 MEP Spaces and Rooms

The Revit MEP model uses both architectural Rooms and MEP Spaces, which are very similar Revit objects that both export as IfcSpace entities, and are both mapped to the COBie Spaces tab by BimServices. In a typical Revit MEP project, Rooms are used to match the Architectural model room layout while Spaces are used for the MEP design, and contain additional MEP properties for analysis. While both Rooms and Spaces export to IFC as IfcSpace entities, only the Room object is used to establish containment relationships for objects in the model.

Model objects must be located within a Room element to report their location in the exported IFC file. Any objects not located within a room, such as pipes and valves run inside walls or below grade, will not report any location information. In an MEP file this can include MEP objects run in the plenum space, since the Room objects are typically constrained to the height of the occupied space.

In order to maximize the number of model objects that correctly report their location, the Room elements in the MEP model have been set to match the full height of each building storey. Additionally, the Ceiling elements in the Architectural model have been set to non-room bounding by de-selecting the Room Bounding property for each element. This allows the Room objects to extend above the ceiling and capture any MEP elements in the plenum space. Finally, Rooms have been added to define the roof areas as well in order to capture any roof top equipment.

While this change to the Room objects will increase the number of model elements that correctly report their location in the IFC export, it will also change the reported height of each Room. Instead of reporting the room height based on usable ceiling height, these Room objects will now report the full height of the building storey. This change is only used on Rooms in the MEP model. The MEP Spaces in the MEP model and the Rooms in the Architectural model are set to match the ceiling height, and report the usable height correctly.

4.8 IFC File Export

The clinic files are provided in their native Revit format, along with an IFC export of each file and the COBie spreadsheet created from each file. IFC files of each model were exported from Revit using the Revit Application Menu > Export > IFC command. The IFC export was created using IFC export settings defined in the *IFC-exportlayers.txt* file included in the common object library.

In order for each of the custom data parameters to export to IFC, each parameter must be filled in. A proprietary custom API routine was used to fill each model object's COBie parameters with text reporting the parameter name. In a live project, this data would be filled in with the appropriate COBie data instead. In addition to the automated routine, the COBie Zone properties applied to the Rooms were manually edited to group the rooms into different Zones. These properties are picked up by the BimServices utility on the COBie Zone tab.

4.9 IFC Model Partitions

The Revit models were initially developed using a single Revit file for the architectural discipline, the structural discipline, and another single Revit file for the MEP/FP disciplines. This allows the designer to take advantage

of Revit's unique capabilities, including parametric connections between objects and model-based analysis or reporting features.

When processing the IFC files using the BimServices Transform1 utility, any IFC file larger than 25 MB caused an "Out of Memory" error and failed to run correctly. In order to create IFC files of less than 25 MB, the Structural and MEP files were divided into partitions based on discipline and building area. A new Revit file was created for each defined area, and all model objects outside of the target area were deleted.

In addition to partitioning the models as described above, the resulting IFC export files were compacted using Solibri IFC Optimizer, a free utility from the authors of Solibri Model Checker. It is described as a lossless IFC optimizer that purges redundant data from the IFC file. In testing the utility on files for this project, Solibri IFC Optimizer reduced the IFC file sizes by 30% to 40% compared to the original Revit export. Note that the use of Solibri IFC Optimizer had an effect on the BimServices COBie transform, as described in section 4.12 of this chapter.

In order to facilitate future updates to the clinic models, the full discipline models are included in the BIM file set, while the partitioned models were created only for the sake of meeting the file size limitations of the BimServices Transform1 utility.

4.10 Limitations of Model Partitions

One drawback of partitioning the Revit model in this manner is that the parametric connections between objects are limited by the file structure. In a typical Revit MEP model, the MEP elements are assigned to "systems" that are used for model analysis and reporting, such as hot and cold water systems or electrical circuits. However, this feature does not work across separate files, so the systems will be incomplete.

For example, a mechanical air diffuser should be part of the mechanical supply or return air system, and be parametrically tied to the air handling unit for that system. Likewise, the air handler should have electrical load requirements, and be tied to the electrical system. When the models are partitioned into separate files, the system connections are broken and the resulting IFC exports do not contain complete MEP system information.

4.11 Export IFC Using “Current View Only” Option

The IFC Export dialog box within Revit has a check box option to export the current view only, instead of the full model. This option allows the user to hide unwanted objects in the current Revit view, and export only the remaining model elements. This will allow the user to setup a 3D view within the overall Revit model showing just the desired area or objects to export. Using this option will avoid the MEP system issue described above, since all of the MEP elements are still contained within the same Revit file. However, when this option is selected, Revit does not export any Room or Space elements (exported as IfcSpace entities), because they are not visible in 3D views within Revit. As a result, the COBie files produced from these IFC exports will not have any entries on the Space tab, and the Component entries will not reference any containing spaces.

4.12 COBie File Conversion

Once the IFC files have been exported from Revit, they are translated to COBie format using the BimServices Transform1 utility along with the `_asCOBIE2.xml.xsl` file. This tool will generate an IFCxml file and a COBie 2.40 spreadsheet from the IFC file. This project used the “all=yes” command line switch in the BimServices utility, which includes all building objects in the COBie output as noted in the program’s documentation.

Each of the COBie spreadsheets was manually checked to verify that the relevant tabs had been correctly filled in. While most of the model objects were correctly translated to the COBie spreadsheet, there were a few exceptions, as noted below.

Missing Data on the Contact Tab: The only Contact tab information available by default in the Revit IFC export is the user name of the person who exported the file, listed as an IfcPerson. BimServices uses this information to create a single entry on the Contact tab and creates an auto-generated email address. Since there is no additional user data in the Revit IFC file, the remaining fields on the Contact tab are left as ‘n/a’ and must be manually filled in.

Missing Data on the Facility Tab: The following fields are not available by default in the Revit IFC export: SiteName, CurrencyUnit, Description, and SiteDescription. These fields must be manually filled in.

Element Type not Exported to IFC: Some object categories within Revit export to IFC without a Type definition. The categories include Revit system family elements that are generally large assemblies or parts of the building such as walls, floors, and roofs. BimServices handles these object types in two different ways:

Create Component Entry without Type: A few object categories are loaded correctly onto the Component tab, but BimServices does not create a Type for these categories.

Create Type as IfcMaterial: BimServices will read the IFC data for certain object categories and create an IfcMaterial entry on the Type tab for these objects.

The Revit categories and IFC export settings are given below, along with a note indicating whether BimServices will create an IfcMaterial entry for the category or load the Component without creating the Type entry.

Table 4-3: Revit Families with no IFC Type.

Revit Category	Export Category	Created IfcMaterial
Ceiling	IfcCovering	Yes
Floor	IfcSlab	Yes
Railing	IfcRailing	No
Roof	IfcRoof, IfcSlab	Yes
Stair	IfcStair, IfcStairFlight, IfcMember	No
Structural Foundation	IfcFooting	No
Structural Framing	IfcBeam	No
Wall	IfcWall	Yes

Type Tab Entry as IfcMaterial: In addition to creating IfcMaterial entries on the Type tab for Ceiling, Floor, Roof, and Wall categories as listed above, BimServices will also load Type tab entries for IfcMaterial entities defined in the IFC file. Since these are materials used in the Revit model, rather than actual model objects, they do not have the same Type data available, nor do they have Component tab entries. These materials are used on the Assembly tab to create assemblies for the Ceiling, Floor, Roof, and Wall categories noted above.

Component Tab Space Field Missing: Some of the Component tab entries are missing data in the Space field. In most cases, this is because the object in the Revit model is not contained within a Room and therefore the IFC export does not include any containment relationship for these objects. Examples include piping that is run within a wall or foundation walls below grade, because they do not contact a Room in the Revit model.

Stair and Railing elements are also missing the Space field, even though these objects are contained within a Room in the Revit model. When these objects are exported from the Revit model to an IFC file, the export does not include the containment relationship.

Missing Attribute tab properties: The Attribute tab includes additional properties associated with the entities in the IFC file. However, some of the Revit object categories were not picked up on the Attribute tab. The Attribute tab does not include any properties for the object categories that do not have a Type defined in the Revit IFC export file (Ceiling, Floor, Railing, Roof, Stair, Structural Foundation, Structural Framing, and Wall categories).

Missing Zone Information: As noted in section 4.8 of this chapter, each of the exported IFC files was compacted using Solibri IFC Optimizer. As a result of the optimization, BimServices no longer processes all of the ZoneName properties that were defined in the Revit model. In the optimized IFC file, BimServices only recognizes the first space associated with each zone, and creates only a single entry on the Zone tab instead of listing every space that is associated with each zone.

A manual review of the optimized IFC file shows that the ZoneName data is still present, but it has been reformatted by Solibri IFC Optimizer and is no longer fully processed by BimServices. One of the ways that Solibri IFC Optimizer reduces the size of an IFC file is by reducing the number of redundant entries used to define an IfcPropertySet. This change is not picked up by the BimServices Transform1 utility when creating the COBie file.

Reviewing a COBie file produced from a non-optimized IFC export against the COBie file produced from the optimized IFC file shows that the Zone tab is the only one affected by the IFC optimization. All of the other COBie

tabs list the same number of entries as the non-optimized file, and data translation for the COBie fields appears to be consistent.

Model Elements Duplicated Across Discipline Files: Certain elements within a Revit model must be duplicated across the separate discipline files in order for the models to function properly. Project data, Grids, Levels, and Rooms are the most common examples.

When the models are created separately and exported to IFC, these elements will be included in each IFC export, using different unique identifiers (GUIDs), even though they are intended to represent the same object within the overall project. This duplication must be taken into account when using the resulting IFC or COBie files for downstream use.

Scheduled Equipment Naming: The government provided documentation for the Clinic building included design drawings for the original facility. These drawings included equipment schedules for everything from the architectural door schedule to the mechanical equipment schedules. In a typical Revit project, each of these items may be named using a different data property, which is used to tag the plans and organize the schedule.

For example, the architectural doors are typically given a unique name for each door element using the “Mark” instance parameter, which is unique to each individual door. Plumbing fixtures are tagged and scheduled by Type instead, with all fixtures of the same type sharing the same designation using the “Type Mark” parameter. Some electrical equipment is organized by the “Panel Name” instance parameter. Other equipment types may use other unique parameters to identify each piece of equipment.

When Revit exports a model to IFC, it assigns “Family Name:Family Type:Revit ID Number” as the unique entity name in the IFC file. BimServices will map this to the Component: Name field in the resulting COBie file. Revit does not directly translate the scheduled name into the IFC entity name field.

In order to consistently map the scheduled object names to the COBie format, the scheduled equipment designations were manually input into

the “TagNumber” parameter, which is picked up on the COBie Component tab.

4.13 COBie File Issues Review

After creating a COBie file from the IFC export, the COBie files were tested using the BimServices Transform1 utility to convert the COBie file back to IFC using the *_fromCOBIE2.ifcxml.xsl* file and the *_Issues.xhtml.xsl* file. This process reviews the COBie file and creates an Issues report in XHTML file format. These results are included with the clinic model files.

When running the *_fromCOBIE2.ifcxml.xsl* file to create an IFC file to test, the IfcRoof entity creates an error that results in an invalid IFC file. Since the *_fromCOBIE2.ifcxml.xsl* file will create both an IFC file and an IFCXML file, the IFCXML file was used to produce the Issues report.

The results of the Issues report show that the COBie files produced Compliance or Adequate Compliance, with the exceptions noted in section 4.12 of this chapter.

4.14 Manual Review of COBie Files

In addition to the BimServices review, the COBie files produced by the BimServices utility were manually reviewed against the objects in the Revit model. To facilitate this review, multi-category schedules were created in Revit to report the objects and data that are exported to IFC and used by BimServices to create the COBie files. Multi-category schedules allow Revit to include objects from multiple categories in a single schedule. This can simplify comparisons to the COBie spreadsheets.

The schedules are included in the native Revit model files, as well as in the common object library templates as Excel files. Note that the schedules produced within Revit are not exact duplicates of the spreadsheets produced by the BimServices utility due to some formatting changes in the file conversion process, but they list the same properties read by the BimServices utility when creating the COBie spreadsheets.

In order to compare the Revit schedules to the COBie spreadsheets, each schedule was exported from Revit and opened in Microsoft Excel. The resulting Excel files are included with the clinic files.

4.15 Structural Model Review

The Structural COBie file *Clinic_S_20110715_asCOBIE2.xls* produced by BimServices was compared to the multi-category schedules produced in the *Clinic_S.rvt* Revit file and their corresponding Excel file exports. Due to limitations of the Revit multi-category schedule, which does not include some system family categories like Floors, Roofs, and Ceilings, the COBie spreadsheet included model objects that are not listed in the Revit multi-category schedules.

Comparison of the objects and data available in the Revit schedule to the spreadsheet data in the COBie file shows that the data translation through IFC is consistent, with the exceptions noted in section 5 of this chapter.

There are 1,078 model objects in the Revit structural model, while the COBie spreadsheet lists 1,080 objects on the Component sheet. Further inspection of the COBie file shows that Stair objects produce multiple entries in the COBie spreadsheet, while they are treated as a single object within Revit. The Stair object is exported to IFC with separate entities for the IfcStair (overall stair object), IfcStairFlight (individual flights), and IfcMember (stringers). This results in separate Component entities for each of these objects, while the Revit file contains just a single Stair object.

With the different structure of the Stair objects taken into account, the number of entities listed on the COBie Component spreadsheet matches the number of entities in the original Revit model.

4.16 Architectural Model Review

The Architectural COBie file *Clinic_A_20110715_asCOBIE2.xls* produced by BimServices was compared to the multi-category schedules produced in the *Clinic_A.rvt* Revit file and their corresponding Excel file exports. Due to limitations of the Revit multi-category schedule, which does not include some system family categories like Floors, Roofs, and Ceilings, the COBie spreadsheet included model objects that are not listed in the Revit multi-category schedules.

Comparison of the objects and data available in the Revit schedule to the spreadsheet data in the COBie file shows that the data translation through IFC is consistent, with the exceptions noted in section 5 of this chapter.

Direct comparison to the Revit file object quantities shows that there are 2,597 objects in the Revit Architecture file, while the COBie spreadsheet lists 2,605 objects on the Component sheet. Further inspection of the COBie file shows that Stair objects produce multiple entries in the COBie spreadsheet, while they are treated as a single object within Revit. The Stair object is exported to IFC with separate entities for the IfcStair (overall stair object), IfcStairFlight (individual flights), and IfcMember (stringers). This results in separate Component entities for each of these objects, while the Revit file contains just a single Stair object.

With the different structure of the Stair objects taken into account, the number of entities listed on the COBie Component spreadsheet matches the number of entities in the original Revit model.

Review of the COBie Type tab shows that most Components are properly related to a Type tab entry, with the exception of the categories noted in section 5 of this chapter. This can also be seen on the Component tab, where the TypeName field for these objects is listed as “n/a”. These object types would need to be manually created to complete the COBie Type tab, or BimServices would need to create Type entries from the component data.

4.17 MEP Model Review

The MEP COBie file *Clinic_MEП_20110715_asCOBIE2.xls* produced by BimServices was compared to the multi-category schedules produced in the *Clinic_MEП.rvt* Revit file and their corresponding Excel file exports. Elements in the MEP object categories are not subject to the limitations of the multi-category schedules found in the architectural file; all of the MEP objects are included in the multi-category schedules.

Comparison of the objects and data available in the Revit schedule to the spreadsheet data in the COBie file shows that the data translation through IFC is consistent, with the exceptions noted in section 4.12 of this chapter.

There are 16,585 total model objects in all of the Revit MEP model partitions, and there are also 16,585 total entities listed on the COBie spreadsheet Component tabs produced by BimServices. The Component list includes all of the objects used in the original Revit model partitions. There are 15,594 model objects in the complete MEP model, since many objects span multiple building areas.

Review of the COBie Type tabs shows that the MEP Components are correctly related to a Type tab entry. However, many of the Revit MEP system families are exported to IFC format with a unique Type entry created for every Component. This would need to be manually corrected in the COBie file, or BimServices would need to consolidate the similar Type entries.

The affected Revit categories are listed in the table below.

Table 4-4: MEP System Families with Multiple COBie Types.

Revit Category	Export Category
Conduits	IfcCableSegmentType
Pipes	IfcPipeSegmentType
Ducts	IfcDuctSegmentType

5 Office Model Description

5.1 Objectives

This chapter describes the creation and testing of BIM models for the office building.

5.2 Approach

The office building was modeled in Revit Architecture 2011 and Revit MEP 2011, based on the government-provided IFC model for the office prototype. The models were exported to IFC format, and the BimServices Transform1 utility was used to create a COBie spreadsheet for each model. The BimServices utility was also used to test the models for compliance using the BimServices Issues tool.

5.3 Scope

The scope of this chapter is to document the creation and testing of BIM files for the office building type. It will also be a reference for future modeling projects.

5.4 Modeling Standards

Model files for the office building include three linked files: one contains the Architectural elements, one contains the Structural elements, and one contains the MEP/FP elements. Additionally, a reference file was created by importing the government-furnished IFC file. This reference file is not part of the Revit BIM file set, but it was linked in and used as a reference to start the new models.

The office BIM consists of the following files:

Table 5-1: Office BIM Files .

File Name	Discipline
Office_A.rvt	Architectural
Office_S.rvt	Structural
Office_MEP.rvt	MEP/F Combined
Office_A_IFC-Import.rvt	For reference only

In addition to the model files listed above, it was necessary to divide the MEP model into smaller files. When processing the IFC files using the BimServices Transform1 utility, any IFC file larger than approximately 25 MB (depending on the model content) caused an “Out of Memory” error and failed to run correctly. In order to create IFC files of less than 25 MB, the MEP models were divided into partitions based on discipline. This process is described in section 5.7 of this chapter.

The MEP models were divided into the following partitions:

Table 5-2: Office MEP Model Partitions .

File Name	Description
Office_ME.rvt	Mechanical, Electrical
Office_PF.rvt	Plumbing, Fire Protection

Note that these partition files are not intended to replace the *Office_ME.rvt* file from the main BIM file list. Some Revit functionality is lost when the building systems are created in separate files, as described in section 5.7 of this chapter.

5.5 File Origin

All BIM files for the office building use the intersection of column grids A/1 as the file origin point. When linking models together, the models will align when the Import Positioning option is set to Auto – Origin to Origin in the Import/Link Revit dialog box.

5.6 File Setup

5.6.1 Reference Model

The layout for the office building is based on a government-furnished model provided in IFC format, which was used as the primary reference for developing the office models. The reference model contained basic architectural elements, but did not include the building structure or MEP/FP systems.

In order to use the IFC file as reference within Revit, the file was imported into a Revit Architecture project file. The imported building was then moved to establish the intersection of column grid A/1 as the origin point, and the file was saved as a new Revit project named *Office_A_IFC-Import.rvt*.

5.6.2 Structural Model

The structural model was created in Revit Architecture using the *Template_A.rvt* Revit template file. The *Office_A_IFC-Import.rvt* reference model was linked in to the Revit file and moved to align column grid A/1 with the Revit file origin point. The linked file was used as a reference to create the structural model, including column grids, foundations, framing, and slabs, and was removed from the file after the model was completed.

After the architectural and MEP models were developed, they were linked into the structural model for coordination. Using the elements in the architectural and MEP models as reference, the structural elements were updated to coordinate across the files. For example, the openings and edges of the floor and roof slabs were updated to match the stair and shaft openings in the architectural model.

The structural model was developed in Revit Architecture 2011 using the components included in the Common Object Library. Note that in a live project, the structural model would be created in Revit Structure, which includes additional tools for analysis and documentation of structural elements.

5.6.3 Architectural Model

The architectural model was created in Revit Architecture using the *Template_A.rvt* Revit template file. The *Office_A_IFC-Import.rvt* reference model was linked in to the Revit file and moved to align column grid A/1 with the Revit file origin point. The linked file was used as a reference to create the architectural model, and was removed from the file after the model was completed.

In addition to linking the reference model into the architectural model file, the *Office_S.rvt* Revit model was also linked into the architectural file, using the Origin to Origin import position setting, to provide the structural elements. Using Revit's Copy/Monitor tool, the grids and levels from the structural model were copied into the architectural model. This ensures that the datum elements (grids and levels) remain consistent across the files, reducing the need for manual adjustment and eliminating potential inconsistencies.

After the MEP model was developed, it was linked into the architectural model for coordination. Using the elements in the MEP model as reference, the architectural elements were updated to coordinate across the files. For example, the furring walls at downspouts and shafts were adjusted to match the MEP layout.

The architectural model was developed in Revit Architecture 2011 using the components included in the Common Object Library.

5.6.4 MEP Model

The MEP model was created in Revit MEP using the *Template_MEPA.rvt* Revit template file. The *Office_A.rvt* and the *Office_S.rvt* architectural and structural models were linked into the new file, using the Origin to Origin import position setting, to be used as reference when creating the MEP model. Using Revit's Copy/Monitor tool, the grids and levels from the structural model were copied into the architectural model.

The MEP model was developed in Revit MEP using the components included in the Common Object Library.

The Revit MEP model uses both Rooms and MEP Spaces, which are very similar Revit objects that both export as IfcSpace entities, and are both

mapped to the COBie Spaces tab by BimServices. In a typical Revit MEP project, Rooms are used to match the Architectural model room layout while Spaces are used for the MEP design, and contain additional MEP properties for analysis. While both Rooms and Spaces export to IFC as IfcSpace entities, only the Room object is used to establish containment relationships for objects in the Revit model.

5.7 IFC File Export

The office files are provided in their native Revit format, along with an IFC export of each file and the COBie spreadsheet created from each file. IFC files of each model were exported from Revit using the Revit Application Menu > Export > IFC command. The IFC export was created using IFC export settings defined in the *IFC-exportlayers.txt* file included in the common object library.

In order for each of the custom data parameters to export to IFC, each parameter must be filled in. A proprietary custom API routine was used to fill each model object's COBie parameters with text reporting the parameter name. In a live project, this data would be filled in with the appropriate COBie data instead. In addition to the automated routine, the COBie Zone properties applied to the Rooms were manually edited to group the rooms into different Zones. These properties are picked up by the BimServices utility on the COBie Zone tab.

5.8 IFC Model Partitions

The Revit models were initially developed using a single Revit file each for the architectural discipline, the structural discipline, and another single Revit file for the MEP/FP disciplines combined. This allows the designer to take advantage of Revit's unique capabilities, including parametric connections between objects and model-based analysis or reporting features.

When processing the IFC files using the BimServices Transform1 utility, any IFC file larger than 25 MB caused an "Out of Memory" error and failed to run correctly. In order to create IFC files of less than 25 MB, the MEP file was divided into partitions based on discipline. A new Revit file was created for each set of disciplines as noted in section 5.4 of this chapter, and all other model objects were deleted.

In addition to partitioning the models as described above, the resulting IFC export files were compacted using Solibri IFC Optimizer, a free utility from the authors of Solibri Model Checker. It is described as a lossless IFC optimizer that purges redundant data from the IFC file. In testing the utility on files for this project, Solibri IFC Optimizer reduced the IFC file sizes by 30% to 40% compared to the original Revit export. Note that the use of Solibri IFC Optimizer had an effect on the BimServices COBie transform, as described in section 5.11 of this chapter.

In order to facilitate future updates to the office models, the full discipline models are included in the BIM file set, while the partitioned models were created only for the sake of meeting the file size limitations of the BimServices Transform1 utility.

5.9 Limitations of Model Partitions

One drawback of partitioning the Revit model in this manner is that the parametric connections between objects are limited by the file structure. In a typical Revit MEP model, the MEP elements are assigned to “systems” that are used for model analysis and reporting, such as hot and cold water systems or electrical circuits. However, this feature does not work across separate files, so the systems will be incomplete.

For example, a mechanical air diffuser should be part of the mechanical supply or return air system, and be parametrically tied to the air handling unit for that system. Likewise, the air handler should have electrical load requirements, and be tied to the electrical system. When the models are partitioned into separate files, the system connections are broken and the resulting IFC exports do not contain complete MEP system information.

5.10 Export IFC Using “Current View Only” Option

The IFC Export dialog box within Revit has a check box option to export the current view only, instead of the full model. This option allows the user to hide unwanted objects in the current Revit view, and export only the remaining model elements. This will allow the user to setup a 3D view within the overall Revit model showing just the desired area or objects to export. Using this option will avoid the MEP system issue described above, since all of the MEP elements are still contained within the same Revit file. However, when this option is selected, Revit does not export any Room or Space elements (exported as IfcSpace entities), because they are not visible

in 3D views within Revit. As a result, the COBie files produced from these IFC exports will not have any entries on the Space tab, and the Component entries will not reference any containing spaces.

5.11 COBie File Conversion

Once the IFC files have been exported from Revit, they are translated to COBie format using the BimServices Transform1 utility along with the `_asCOBIE2.xml.xsl` file. This tool will generate an IFCxml file and a COBie 2.40 spreadsheet from the IFC file. This project used the “all=yes” command line switch in the BimServices utility, which includes all building objects in the COBie output as noted in the program’s documentation.

Each of the COBie spreadsheets was manually checked to verify that the relevant tabs had been correctly filled in. While most of the model objects were correctly translated to the COBie spreadsheet, there were a few exceptions, as noted below.

Missing Data on the Contact Tab: The only Contact tab information available by default in the Revit IFC export is the user name of the person who exported the file, listed as an IfcPerson. BimServices uses this information to create a single entry on the Contact tab and creates an auto-generated email address. Since there is no additional user data in the Revit IFC file, the remaining fields on the Contact tab are left as ‘n/a’ and must be manually filled in.

Missing Data on the Facility Tab: The following fields are not available by default in the Revit IFC export: SiteName, CurrencyUnit, Description, and SiteDescription. These fields must be manually filled in.

Element Type not Exported to IFC: Some object categories within Revit export to IFC without a Type definition. The categories include Revit system family elements that are generally large assemblies or parts of the building such as walls, floors, and roofs. BimServices handles these object types in two different ways:

Create Component Entry without Type: A few object categories are loaded correctly onto the Component tab, but BimServices does not create a Type for these categories.

Create Type as IfcMaterial: BimServices will read the IFC data for certain object categories and create an IfcMaterial entry on the Type tab for these objects.

The Revit categories and IFC export settings are given below, along with a note indicating whether BimServices will create an IfcMaterial entry for the category or load the Component without creating the Type entry.

Table 5-3: Revit Families with no IFC Type.

Revit Category	Export Category	Created IfcMaterial
Ceiling	IfcCovering	Yes
Floor	IfcSlab	Yes
Railing	IfcRailing	No
Roof	IfcRoof, IfcSlab	Yes
Stair	IfcStair, IfcStairFlight, IfcMember	No
Structural Foundation	IfcFooting	No
Structural Framing	IfcBeam	No
Wall	IfcWall	Yes

Type Tab Entry as IfcMaterial: In addition to creating IfcMaterial entries on the Type tab for Ceiling, Floor, Roof, and Wall categories as listed above, BimServices will also load Type tab entries for IfcMaterial entities defined in the IFC file. Since these are materials used in the Revit model, rather than actual model objects, they do not have the same Type data available, nor do they have Component tab entries. These materials are used on the Assembly tab to create assemblies for the Ceiling, Floor, Roof, and Wall categories noted above.

Component Tab Space Field Missing: Some of the Component tab entries are missing data in the Space field. In most cases, this is because the object in the Revit model is not contained within a Room and therefore the IFC export does not include any containment relationship for these objects. Examples include piping that is run within a wall or foundation walls below grade, because they do not contact a Room in the Revit model.

Stair and Railing elements are also missing the Space field, even though these objects are contained within a Room in the Revit model. When these

objects are exported from the Revit model to an IFC file, the export does not include the containment relationship.

Missing Attribute tab properties: The Attribute tab includes additional properties associated with the entities in the IFC file. However, some of the Revit object categories were not picked up on the Attribute tab. The Attribute tab does not include any properties for the object categories that do not have a Type defined in the Revit IFC export file, as listed above in Table 5-3.

Missing Zone Information: As noted in section 5.7 of this chapter, each of the exported IFC files was compacted using Solibri IFC Optimizer. As a result of the optimization, BimServices no longer processes all of the ZoneName properties that were defined in the Revit model. In the optimized IFC file, BimServices only recognizes the first space associated with each zone, and creates only a single entry on the Zone tab instead of listing every space that is associated with each zone.

A manual review of the optimized IFC file shows that the ZoneName data is still present, but it has been reformatted by Solibri IFC Optimizer and is no longer fully processed by BimServices. One of the ways that Solibri IFC Optimizer reduces the size of an IFC file is by reducing the number of redundant entries used to define an IfcPropertySet. This change is not picked up by the BimServices Transform1 utility when creating the COBie file.

Reviewing a COBie file produced from a non-optimized IFC export against the COBie file produced from the optimized IFC file shows that the Zone tab is the only one affected by the IFC optimization. All of the other COBie tabs list the same number of entries as the non-optimized file, and data translation for the COBie fields appears to be consistent.

Model Elements Duplicated Across Discipline Files: Certain elements within a Revit model must be duplicated across the separate discipline files in order for the models to function properly. Project data, Grids, Levels, and Rooms are the most common examples.

When the models are created separately and exported to IFC, these elements will be included in each IFC export, using different unique identifiers (GUIDs), even though they are intended to represent the same

object within the overall project. This duplication must be taken into account when using the resulting IFC or COBie files for downstream use.

Scheduled Equipment Naming: In a typical Revit project, each of the scheduled equipment items may be named using a different data property, which is used to tag the plans and organize the schedules. For example, the architectural doors are typically given a unique name for each door element using the “Mark” instance parameter, which is unique to each individual door. Plumbing fixtures are tagged and scheduled by Type instead, with all fixtures of the same type sharing the same designation using the “Type Mark” parameter. Other equipment types may use other unique parameters to identify each piece of equipment.

When Revit exports a model to IFC, it assigns “Family Name:Family Type:Revit ID Number” as the unique entity name in the IFC file. BimServices will map this to the Component: Name field in the resulting COBie file. Revit does not directly translate the scheduled name into the IFC entity name field. In order to consistently map the scheduled object names to the COBie format, the scheduled equipment designations must be input into the “TagNumber” parameter, which is picked up on the COBie Component tab.

5.12 COBie File Issues Review

After creating a COBie file from the IFC export, the COBie files were tested using the BimServices Transform1 utility to convert the COBie file back to IFC using the *_fromCOBIE2.ifcxml.xsl* file and the *_Issues.xhtml.xsl* file. This process reviews the COBie file and creates an Issues report in XHTML file format. These results are included with the office model files.

When running the *_fromCOBIE2.ifcxml.xsl* file to create an IFC file to test, the IfcRoof entity creates an error that results in an invalid IFC file. Since the *_fromCOBIE2.ifcxml.xsl* file will create both an IFC file and an IFCXML file, the IFCXML file was used to produce the Issues report.

The results of the Issues report show that the COBie files produced Compliance or Adequate Compliance, with the exceptions noted in section 5.11 of this chapter.

5.13 Manual Review of COBie Files

In addition to the BimServices review, the COBie files produced by the BimServices utility were manually reviewed against the objects in the Revit model. To facilitate this review, multi-category schedules were created in Revit to report the objects and data that are exported to IFC and used by BimServices to create the COBie files. Multi-category schedules allow Revit to include objects from multiple categories in a single schedule. This can simplify comparisons to the COBie spreadsheets.

The schedules are included in the native Revit model files, as well as in the common object library templates as Excel files. Note that the schedules produced within Revit are not exact duplicates of the spreadsheets produced by the BimServices utility due to some formatting changes in the file conversion process, but they list the same properties read by the BimServices utility when creating the COBie spreadsheets.

In order to compare the Revit schedules to the COBie spreadsheets, each schedule was exported from Revit and opened in Microsoft Excel. The resulting Excel files are included with the clinic files.

5.14 Structural Model Review

The Structural COBie file *Office_S_20110811_asCOBIE2.xls* produced by BimServices was compared to the multi-category schedules produced in the *Office_S.rvt* Revit file and their corresponding Excel file exports. Due to limitations of the Revit multi-category schedule, which does not include some system family categories like Floors, Roofs, and Ceilings, the COBie spreadsheet included model objects that are not listed in the Revit multi-category schedules.

Comparison of the objects and data available in the Revit schedule to the spreadsheet data in the COBie file shows that the data translation through IFC is consistent, with the exceptions noted in section 5 of this chapter.

There are 494 model objects in the Revit structural model, while the COBie spreadsheet lists 494 objects on the Component sheet. The number of entities listed on the COBie Component spreadsheet matches the number of entities in the original Revit model.

5.15 Architectural Model Review

The Architectural COBie file *Office_A_20110811_asCOBIE2.xls* produced by BimServices was compared to the multi-category schedules produced in the *Office_A.rvt* Revit file and their corresponding Excel file exports. Due to limitations of the Revit multi-category schedule, which does not include some system family categories like Floors, Roofs, and Ceilings, the COBie spreadsheet included model objects that are not listed in the Revit multi-category schedules.

Comparison of the objects and data available in the Revit schedule to the spreadsheet data in the COBie file shows that the data translation through IFC is consistent, with the exceptions noted in section 5.11 of this chapter.

Direct comparison to the Revit file object quantities shows that there are 791 objects in the Revit Architecture file, while the COBie spreadsheet lists 799 objects on the Component sheet. Further inspection of the COBie file shows that Stair objects produce multiple entries in the COBie spreadsheet, while they are treated as a single object within Revit. The Stair object is exported to IFC with separate entities for the IfcStair (overall stair object), IfcStairFlight (individual flights), and IfcMember (stringers). This results in separate Component entities for each of these objects, while the Revit file contains just a single object for each stair element.

With the different structure of the Stair objects taken into account, the number of entities listed on the COBie Component spreadsheet matches the number of entities in the original Revit model.

Review of the COBie Type tab shows that most Components are properly related to a Type tab entry, with the exception of the categories noted in section 5.11 of this chapter. This can also be seen on the Component tab, where the TypeName field for these objects is listed as “n/a”. These object types would need to be manually created to complete the COBie Type tab, or BimServices would need to create Type entries from the component data.

5.16 MEP Model Review

The MEP COBie file *Office_MEPM_20110811_asCOBIE2.xls* produced by BimServices was compared to the multi-category schedules produced in

the *Office_MEП.rvt* Revit file and their corresponding Excel file exports. Elements in the MEP object categories are not subject to the limitations of the multi-category schedules found in the architectural file; all of the MEP objects are included in the multi-category schedules.

Comparison of the objects and data available in the Revit schedule to the spreadsheet data in the COBie file shows that the data translation through IFC is consistent, with the exceptions noted in section 5.11 of this chapter.

There are 5,697 total model objects in the Revit MEP models, and there are also 5,697 total entities listed on the COBie spreadsheet Component tabs produced by BimServices. The Component list includes all of the objects used in the original Revit model partitions.

Review of the COBie Type tabs shows that the MEP Components are correctly related to a Type tab entry. However, many of the Revit MEP system families are exported to IFC format with a unique Type entry created for every Component. This would need to be manually corrected in the COBie file, or BimServices would need to consolidate the similar Type entries.

The affected Revit categories are listed in the table below.

Table 5-4: MEP System Families with Multiple COBie Types.

Revit Category	Export Category
Conduits	IfcCableSegmentType
Pipes	IfcPipeSegmentType
Ducts	IfcDuctSegmentType

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) September 2011	2. REPORT TYPE Final	3. DATES COVERED (From - To)		
4. TITLE AND SUBTITLE Experimental Building Information Models		5a. CONTRACT NUMBER W912HZ-D-0003		
		5b. GRANT NUMBER		
		5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Mark Johnson and Kristine K. Fallon		5d. PROJECT NUMBER		
		5e. TASK NUMBER		
		5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Kristine Fallon Associates, Inc. 11 E. Adams Street, Suite 1100 Chicago, IL 601603		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center Construction Engineering Research Laboratory PO Box 9005 Champaign, IL 61826-9005		10. SPONSOR/MONITOR'S ACRONYM(S) ERDC-CERL		
		11. SPONSOR/MONITOR'S REPORT NUMBER(S) ERDC/CERL CR-11-2		
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT In a previous Army research project, a core life-cycle building information model (BIM) was developed based on example models for three specific buildings: a Duplex Apartment, a Clinic, and an Office. These models were developed inconsistently over time by different modelers and contain a various levels of detail and quality of content across disciplines. This report documents the development and creation of three building information models that include information for architectural, structural, plumbing, electrical, heating, and ventilating. The three buildings are a family housing unit, a small office building, and a medical clinic. The authors developed a set of electronic building components, or common object library, and necessary setup protocols, and created a model-development plan to provide guidance to modelers. These guidelines may be helpful to designers and BIM managers to successfully create models that comply with Industry Foundation Class (IFC) based standards such as the Facility Management Handover Model View Definition or Construction Operations Building information exchange (COBie) format.				
15. SUBJECT TERMS building information modeling (BIM), Industry Foundation Class (IFC), Construction Operations Building Information Exchange (COBie), life-cycle planning, sustainable facilities				
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified	87	19b. TELEPHONE NUMBER (include area code)